Department of Treasury
U.S. Customs Service
Year 2000 Program Office

Operational Program Management Plan

Table of Contents

List of Figures

List of Appendices

1.0 Introduction

The U.S. Customs Service has determined that the majority of its computer systems are, or soon will be, at risk because of their potential inability to process data accurately as the calendar date moves closer to January 1, 2000. The Year 2000 problem is imminent. While many systems will operate until December 31, 1999, some systems will fail earlier. Customs must identify which systems will fail due to the century rollover, when these systems will fail, and how and when these systems must be fixed.

This document provides an operational plan for organizations within Customs that are converting data and/or renovating applications to remove the Year 2000 problem. It describes the overall Customs approach to Year 2000 compliance, specific guidelines for achieving compliance, and requirements of the business system project teams to the Year 2000 Program Office. The Customs Year 2000 Strategic Plan, is a companion document which provides an overview of the Customs Year 2000 renovation approach.

1.1 Purpose

For organizations within Customs that are converting data or renovating applications to Year 2000 compliance, this document details the scope of the Customs Year 2000 program; the goal of the plan, and specific objectives that accomplish the plan goal. This is a management document; it offers guidance to all Customs staff and management, whether federal employee or contract staff, regarding the desired renovation approach to be employed at Customs.   It is the intention of the Year 2000 Program Office to plan and provide oversight of the Year 2000 program at Customs as a single information system development effort, with several subsystems being renovated according to a schedule that meets the overall goals of the Agency.

1.2 Authority

This plan, and the companion Strategic Plan, fully supports the goals and direction of the Treasury Year 2000 Date Conversion Program Management Plan[1], the GAO Assessment Guide[2], and the Customs FY98 Information and Technology Process Annual Plan. In addition, the plan objectives address many of the issues mandated by the Clinger-Cohen Act (ITMRA) of 1996.

[1] Treasury Year 2000 Date Conversion Program Management Plan, Department of the Treasury, Deputy Assistant Secretary for Information Systems and Chief Information Officer, Office of Information Resources Management, December 18, 1996.

[2] Exposure Draft entitled Year 2000 Computing Crisis: An Assessment Guide (GAO/AIMD-10.1.14, February 1997).

This plan has the full endorsement of the Office of Information Technology (OIT), Chief Operating Officer, Robert McNamara, Office of Finance, Chief Financial Officer, Vincette Goerl, and of the Year 2000 Program Management Office, Director, Sharon Mazur. Compliance by the affected OIT Divisions with its directives is expected.

1.3 Document Organization

This plan is organized into two chapters, and several appendices. This, the first chapter, introduces the plan, and identifies the authority for the plan. The second chapter describes the plan in detail, presenting details of the Strategic Plan, which achieve the overall agency plan goal for both information technology (IT) assets, and non-IT assets.

2.0 Operational Plan

2.1 Goal

Ensure that all computer information technology (IT) systems and non-IT systems, currently maintained by Customs, are able to process date requirements that span the century rollover.

The Customs Year 2000 Program Office recognizes that systems of this nature are fundamentally either mission-critical (tier 1), non-mission critical (tier 2), or other (tier 3). Customs intends a phased approach to becoming Year 2000 compliant by addressing all mission critical systems first, followed by non-mission critical systems. Customs consistently defines mission critical systems to be those that cause a loss of Customs core capability when system failure occurs. The mission critical IT systems are currently defined, at the highest level, to be ACS, TECS, and the Administrative systems. The definition of mission-critical non-IT systems is pending. As process analysis and decomposition occurs, a more thorough specification of mission-critical will occur.

The summarized inventories for each of these major mission-critical IT systems are presented in Appendix A, Summarized IT Inventory. A summarized non-IT inventory remains pending.Appendix B, Integrated Mission-Critical Schedule presents a high-level view of the overall program schedule. Major mission critical program milestones are presented in Table 1, below.

Table 1. Customs Major Program Milestones

Program Milestone	Treasury Mandated Completion Date	Customs Actual Completion Date	Milestone Notes
Awareness/Assessment	May 1996	ongoing	This phase will continue throughout the life cycle of the project.
Project/Strategy Plan	June 1996	July 1997	The plan is updated frequently to reflect program updates

Software Inventory	July 1996	August 1997	
Resource Estimates	July 1996	August 1997	
Prioritization and Schedule	August 1996		Estimated completion October 1997
Methodology, Tools and Techniques	September 1996	February 1997	
Conversion Pilot Project	January 1997		This project is in progress.
Strategy and Methodology Adjustment	February 1997	October 1997	
Conversion Implementation and Testing	July 1998		
Contingency Plan	September 1998		Customs believe this date to be too late in the process and will strive towards a Spring 1998 date.
Training/Communication	Ongoing Effort		

2.2 Customs Guidelines for Achieving Year 2000 Compliance

Objectives A through L are defined to collectively achieve the goal of this Operational Plan. These objectives are not listed in any chronological order. A schedule will be prepared to show planned completion dates. It is envisioned that each project team will accomplish their objectives with technical and management support from the Year 2000 Program Office, when necessary. Each objective is supported by specific strategies. This section itemizes each strategy and offers operational guidance for implementing the strategies.

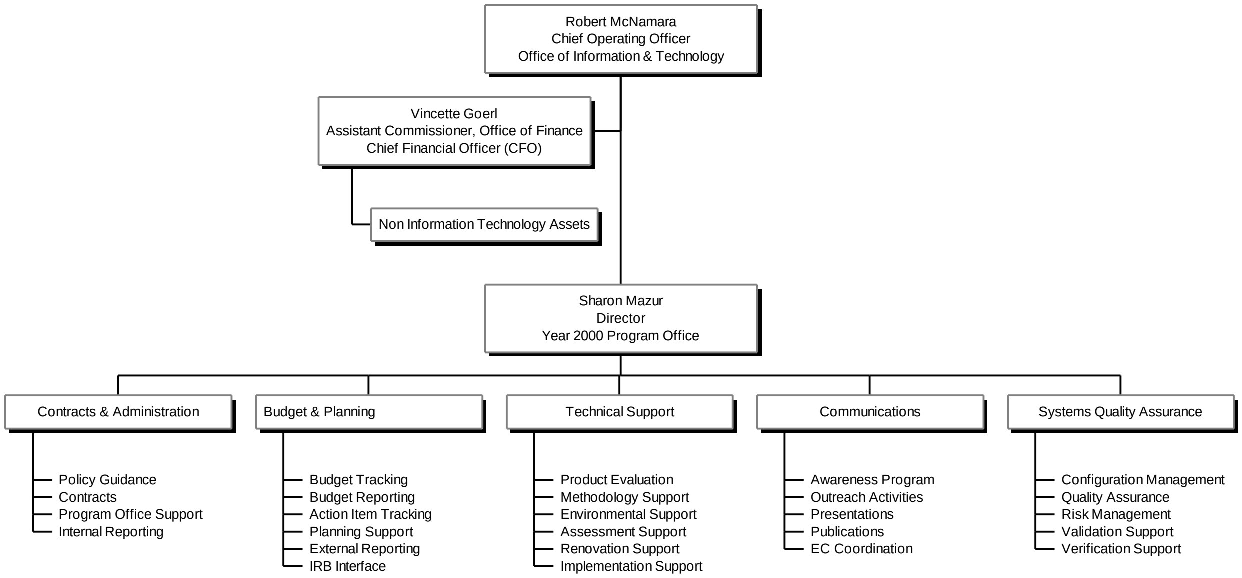
Objective A. Centrally Manage The Year 2000 Program

The Year 2000 program is an agency problem and is a survival issue for the Customs Service. This is the single largest and most complex system conversion effort undertaken by Customs. It requires the disciplined and coordinated application of scarce resources to an enterprise-wide system conversion effort that must be complete by a fixed date. To succeed, Customs must centrally manage the Year 2000 program as a large system development effort.

Strategies:

A.a. Establish a Year 2000 Program Office

The Year 2000 Program Office is dedicated to the successful conversion of Customs IT and non-IT assets. The chart presented in Figure 1 depicts the Year 2000 Program Office hierarchy and its broad responsibilities. In general, the Year 2000 Program Office has the responsibility for coordinating and providing oversight to all activities relating to the renovation of computer programs and equipment containing date sensitive chips to ensure that they are fully Year 2000 compliant. The office is divided into 5 teams, each responsible for different aspects of the compliance effort. It is the mission of the Year 2000 Program Office to provide technical support and management oversight to the project teams in their effort to bring their systems into a state of Year 2000 compliance. Relationships exist between the Year 2000 Program Office, the project teams, the quality assurance teams, and ultimately, the process owner.



Specific Actions:

1. Segregate Year 2000 activities into branches.
2. Determine resources required.
3. Develop method (plan) to obtain resources.
4. Establish Year 2000 program manager.
5. Establish team leads.
6. Obtain resources.
7. Develop a budget.
8. Develop a charter.
9. Have charter approved by COO & CFO.

A.b. <u>Establish an Executive Council</u>

An Executive Council (EC) was established to provide continual guidance and policy direction and approval among the programmatic and functional area managers on priorities and mission impact. The Executive Council also provides quick resolution when conflicts arise. The EC is co-chaired by the Chief Operating Officer and the Chief Financial Officer, and is comprised of project team managers, process owners, and business area owners. The EC can establish ad hoc subordinate groups, which convene, as necessary to fill special purpose requirements.

Specific Actions:

1. **Create charter.**
2. **Determine membership.**
3. **Approve charter.**
4. **Set up schedule and mechanism for meeting.**

A.c. Establish and Track the Overall Program Plan

This document, the Customs Year 2000 Operational Program Management Plan, serves as the mechanism for establishing the overall program plan. Once baselined, it will be updated periodically to reflect the current progress of the Year 2000 program at Customs. Milestones have been established (Table 1), with estimated target dates for completion, which will be closely monitored for issues and constraints.

Specific Actions:

1. Develop a strategic plan.

Deliverable: Year 2000 Strategic Plan	**Responsible: Year 2000 Program Office**

2. Develop an operational plan.

Deliverable: Year 2000 Operational Plan	**Responsible: Year 2000 Program Office, with input from the Project teams and Stakeholders**

3. Track the progress against the plan (develop procedures to do this).

4. Implement procedures.

A.d. Develop Reporting Requirements

The Department of the Treasury, Office of Information Resources Management (OIRM) has developed specific requirements for each of the Treasury bureaus regarding the reporting of progress towards completion of their respective Year 2000 conversion efforts. To that end, the Year 2000 Program Office has similar requirements of the project teams. Appendix C presents the reporting requirements of the project teams to the Year 2000 Program Office. It is the intention of the Year 2000 Program Office to report progress metrics at the subsystem level for each lifecycle phase (renovation, validation, verification, and implementation). These metrics will be reported weekly or as otherwise directed, to satisfy requirements for reporting to OIT and Treasury. Members of the Year 2000 Program Office will facilitate data collection.

Deliverable: Year 2000 Reporting Requirements Definition	**Responsible: Year 2000 Program Office**

A.e. <u>Establish Performance Measures / Reporting Metrics</u>

The Treasury Department, OIRM has established four performance goals for measuring the performance of its bureaus. The Departmental strategy states that all Treasury systems must be Year 2000 compliant to avoid service disruptions. The Treasury goals are:

S no processing errors due to calendar rollover
S no critical system failures due to date rollover
S all systems converted and validated before fiscal Year 2000
S all mission-critical systems converted by the end of fiscal year 1998

To achieve the Departmental goals, the Customs Year 2000 Program Office must establish performance indicators, then measure progress against the indicators. Performance indicators are categorized as scope, schedule, and resources.

The intent is that the metrics shown in Appendix C are natural products of project management, and that they do not require extraordinary means to collect and report. Initial estimates will be established by the project teams, and progress tracked by the Year 2000 Program Office. Progress and/or deliverables are not to exceed two calendar weeks.

Deliverable: Year 2000 Performance Measures Guideline	**Responsible: Year 2000 Program Office**

Objective B. Ensure Year 2000 Awareness

Ensuring Year 2000 awareness is an objective linked to executive sponsorship. Since Customs has full sponsorship at the executive levels, this objective is focused on communicating the issues to the business units, internal personnel and external business partners.

Strategies:

B.a. <u>Develop an Outreach Plan</u>
The purpose of the Year 2000 Outreach Plan is to define the approach Customs is pursuing to increase awareness of the Year 2000 problem among our internal and external customers, and to foster continuing Customs-wide support and sponsorship.

Deliverable: Year 2000 Outreach Plan	**Responsible: Year 2000 Program Office**

B.b. Implement the Outreach Plan

To successfully promote awareness throughout Customs, it is essential that the Year 2000 Project Team establish a presence by providing oral presentations to Customs personnel at every level. This will offer credibility to their efforts and allow those with questions a chance to ask them directly. This will be accomplished by providing briefings to the following offices by the end of the first quarter in fiscal year 1998.

C Field Operations
C Investigations
C Congressional and Public Affairs
C Regulations and Rulings
C Internal Affairs
C International Affairs
C Program Planning
C Trade Ombudsman
C Chief Council
C Finance
C Human Resources and Management
C Information and Technology

B.c. Maintain a Central Database of Program Artifacts

A Year 2000 library will be created and will contain documents and publications relevant to the Year 2000 challenge. Documents will be scanned and stored in an automated database. Relevant documents will be posted to the Customs Electronic Bulletin Board or the Year 2000 home page on either the Customs Internet or Intranet, whichever is appropriate.

Objective C. Establish the Enterprise Inventory

An enterprise-wide inventory of IT systems and their components, and non-IT assets provides the necessary foundation for Year 2000 program planning. A thorough inventory ensures that all systems are identified and linked to a specific business area, and that all enterprise-wide, cross-boundary system dependencies are considered.

The Customs Year 2000 Program Management Office recognizes that the enterprise inventory contains two fundamental types of inventory that are eligible for renovation: mission-critical and non-mission-critical. Within those two types are information technology (IT) systems, and non-IT systems. Customs intends a phased approach to becoming Year 2000 compliant by addressing IT and non-IT mission-critical systems first, followed by IT and non-IT mission-support systems.

Strategies:

C.a. Define the methodology for inventory collection

The inventories developed to satisfy this objective should be elementary components, meaning that no further decomposition exists. It is important for each project team to approach the inventory process carefully, so that the results are consistent and accurate. The intent is to identify each discrete component eligible for renovation. This then, becomes one measure of renovation scope.

Specific Actions:

1. **Define the method for establishing your inventory**.
   For each of the following inventory types, define the method for establishing the inventory. Define the inventory collection methodology for the following inventory types:
   - Type 1. Customs software components (COTS, application software, hardware, etc.)
   - Type 2. Databases and database tables that support Customs software
   - Type 3. Customs non-IT assets (hardware, phone switches, etc.)

   The method must be repeatable. In all cases, the method must be repeatable (document the process and manage the results).

Required IT and non-IT Deliverable: Inventory Methodology	**Responsible: Project teams** **Support: Year 2000 Program Office**

2. **Baseline each inventory type**.
   Using the methodology defined in Objective C.a.1, baseline each inventory type, and place under configuration management control.

Required IT and non-IT Deliverable: **Baselined Software Inventory** **Baselined Data Inventory** **Baselined Non-IT Inventory**	**Responsible: Project teams** **Support: Year 2000 Program Office**

3. **Conduct a cross-reference analysis.**
   This will give confidence that the inventory contains only relevant objects, that redundant or unused objects are eliminated, and that all objects contain their requisite components. Use of an automated tool or toolset is highly recommended. For example:
   a. For each production JCL member, find the executable program,
   b. For each data definition (//DD...), find the physical file or database,
   c. For each database, find the database schema definition (DBD, DDL, etc.)
   d. For each source member, find the correct load module. Find all static or dynamic links associated to the load module,

e. Find the build procedures,
f. For programs with an external user interface, find the screen or window definition,
g. Find all copybook members.

Update the inventory list to reflect the results of the cross-reference analysis. This analysis concludes with decision to renovate, replace, retire or re-engineer. **If any members are missing**, the inventory must be immediately identified as having an issue. Establish a *Resolution Team* to resolve the missing and unclaimed source code.

Required IT Deliverable: Updated Inventories	**Responsible: Project teams** **Support: Year 2000 Program Office**

C.b. Develop Mechanism for Inventory Management and Tracking

The Year 2000 Program Office will devise a mechanism to facilitate consistent and accurate tracking of the Customs software, hardware, data, and non-IT assets.

Required Deliverable: Progress Reporting Mechanism	**Responsible: Year 2000 Program Office**

C.c. Report Progress Against Inventory Baseline

Monthly, or as otherwise directed, the project teams will report renovation progress against baseline estimates to the Year 2000 Program Office who will, in turn, compile the results for distribution within Customs, and to other external government entities, such as Treasury OIRM and GAO.

Objective D. Plan for Conversion Resources

Due to the complexities and scope of the Year 2000 problem, it is critical that Customs develop comprehensive plans that establish schedules for all tasks and phases of the Year 2000 program, assign conversion projects to teams, anticipate and acquire resources for the effort.

Strategies:

D.a. Identify Mission Critical Assets
Once the enterprise inventories have been baselined, each project team must identify the mission-critical assets. This process assumes that within each of the five Customs major systems (ACS, ACE, AES, TECS and Admin) there will be

some parts of it that are more critical than others. Identification of these mission-critical parts helps to establish priorities, and schedules for the project team.

Required IT and non-IT Deliverable: Mission-Critical Component List	**Responsible: Project teams**

D.b. Prioritize and Schedule Subsystems and Applications

Recognizing the criticalities identified above, each project team must **develop a detailed and prioritized plan of renovation activities**. The plan must address all conversion lifecycle activities (assessment, renovation, validation, and implementation). Once developed, the plan should be baselined on the estimated start and finish dates. The Year 2000 Program Office recommends dividing the renovation workload into large, manageable segments, which are more easily tracked than hundreds of discrete elementary items. **Before baselining the schedule, conduct coordination meetings with the adjoining project teams to work out scheduling conflicts.** This will result in the formation of a Migration plan to include releases of software and corresponding components to production.

Specific Actions:

1. **Create a prioritized list of system components**.
Use a project management tool, such as Microsoft Project, PS7 or an Excel spreadsheet, to create a prioritized list (most critical to least critical) of system components. Be sure to include software, data, and non-IT components.

2. **Identify the baseline scheduling metrics for each component.**
For each component, identify the resource to effect renovation, planned start date, planned completion date, any constraints and dependencies. The deliverable could take the form of a Gantt chart or a spreadsheet.

Required IT and non-IT Deliverable: Prioritized Renovation Schedule	**Responsible: Project teams**

3. **Conduct coordination meetings with the adjoining project teams.** These meetings are to work out scheduling conflicts. Using the inventories established in Objective C.a.3, to identify shared inventories across project teams. Prepare a migration plan to resolve scheduling conflicts, and to level the workload across the fixed timeframe.

Required IT Deliverable: Migration Plan	**Responsible: Project teams**

D.c. Estimate and Acquire Necessary Renovation Resources

Given the known renovation workload requirements revealed by the prioritized renovation schedule, estimate renovation resource shortfalls, and develop a plan to acquire additional resources. The plan must consider the scope of work (specification, conversion, unit testing and documentation), and appropriate lead time (e.g., recruiting, security clearances, and contract vehicles), and logistical considerations.

Specific Actions:

1. **Estimate the project team resources required**.
   Using the results of the Prioritized Renovation Schedule developed in D.b.2, estimate the project team resource requirement.

2. **Develop a plan to acquire additional resources**.
   The Year 2000 Program Office will assist with identifying available sources of resources. The plan should identify required resource type, required resource quantity, the planned acquisition date, and lead time. Include resources in a pending status. This plan must be updated periodically.

Required IT and non-IT Deliverable: Resource Acquisition Plan	**Responsible: Project teams** **Support: Year 2000 Project Office**

Objective E. Determine Renovation Methodology

Customs will follow the GAO guidelines for conversion strategy, and will tailor the guidelines to satisfy specific local requirements. In addition, each project team will develop detailed procedures for ensuring system integrity during the renovation, validation, and implementation phases.

E.a. Develop and Publish Conversion Procedures, Such as Configuration Management and Quality Assurance

Process management is critical to the success of renovation. The Year 2000 Program Office is committed to providing assistance to the project teams in the development and use of process plans. Appendices D and E present Year 2000 Program Office guidance for Quality Assurance (QA) and Configuration Management (CM) plans. Each project team should plan for time to review their existing procedures to identify weaknesses. This objective component should clearly identify the procedures specific to the project team. The plans should be absolutely clear about items such as product baselining, configuration management, and problem tracking procedures, as well as requirements for turnover to the quality assurance teams, and acceptance procedures.

Specific Actions:

1. **Review your existing development and maintenance procedures to identify weakness**. Pay special attention to weaknesses that could easily be strengthened for this effort.

2. **Develop configuration management procedures.**
Using the Year 2000 Program Office CM Plan as a baseline, identify the procedures your team will use to ensure the integrity of renovation changes. This is not limited to software changes. Include data changes, hardware, [user] procedure changes and changes to documentation. For non-IT systems, define the procedures for fixed asset management tracking, identifying what happens when products are replaced.

Required IT and non-IT Deliverable: Configuration Management Procedures	**Responsible:** **Support:**	**Project teams** **Year 2000 Program Office**

3. **Develop quality assurance procedures.**
Using the Year 2000 Program Office QA Plan as a baseline, identify the procedures your team will use to ensure a quality product. For non-IT systems, define the procedures used for ensuring that the renovated products arrive at their proper destinations in a timely manner and are intact.

Required IT and non-IT Deliverable: Quality Assurance Procedures	**Responsible:** **Support:**	**Project teams** **Year 2000 Program Office**

E.b. Determine Analysis and Conversion Tool

This objective component will yield a list of tool candidates that work within the constraints of the project team=s environment. While it is recognized that tools alone cannot solve the remediation problem, it is clear that some tools can improve the quality of the repairs. A variety of tool categories should be considered, such as: intelligent source code parsers, executable code analyzers, control language parsers, data file comparison tools, and test script management tools to name a few.

Specific Actions:

1. **Establish a list of tools.**
Evaluate the commercial and government marketplace for tools that may assist in the conversion effort as well as expedite the process and reduce overall labor and time requirements. Include tools for analysis, conversion,

and testing. Recognize that the use of tools may influence CPU times, DASD, etc.

Required IT Deliverable: List of Tool Candidates	**Responsible: Year 2000 Program Office**

2. **Make recommendations for appropriate approval**.
While it is recognized that there may be a need for one or more special purpose tools, commonality of tools across Customs is desired, to the extent possible.

Optional Deliverable: Tool Recommendations	**Responsible: Project teams** **Support: Year 2000 Program Office**

E.c. Define Year 2000 Compliance Approach *(SDLC Counterpart: Requirements Definition)*

Simply stated, compliance is the ability to accurately process dates that span multiple centuries. But taken further, there are many flexibilities that Customs can apply. **Each project team must decide their approach to becoming compliant, factoring in the Customs definition[3], internal and external constraints.** For example, compliance could be defined as described in Appendix F. Examining the example shows the value of defining compliance for all phases of the effort. When a date field is encountered, team members know exactly how to deal with it. The Appendix F example is not exhaustive; there are many permutations that could be considered acceptable. The definition must, however, be compatible with the definitions of adjoining project teams and external partners. Where the definitions cannot be made compatible, data bridges, which convert from one format to another, must be built and implemented.

Specific Actions:

1. **Review the Compliance Approach Definition example in Appendix F.**

2. **Make tailoring decisions for each category of definition: renovation approach, interface approach, and data approach.** Assuming that the majority of non-IT systems will be replaced if found non-compliant, define the minimum acceptance criteria for compliance. If the non-IT system is to be renovated, define the compliance approach, as if for an IT system.

[3] Customs adopts the definition as published in the Federal Acquisition Register.

Required IT and non-IT Deliverable: Compliance Approach Definition (Appendix F)	Responsible: Project teams

3. **Make note of incompatibilities among groups of software components, and across project teams.**
   Where incompatibilities exist, bridges will need to be built. Also, **consider the interfaces to archived data**. If your archived data *is not* renovated, but the active data *is* renovated, bridges will need to be built. **Identify necessary data bridges.** Identify the applications needing a bridge, and define the bridge requirements. Bridges are new programs in need of development, that must be considered in the Prioritized Renovation Schedule (D.b) for migration to and from bridging processes as appropriate.

Required IT Deliverable: Identify Necessary Bridges	Responsible: Project teams

E.d. Develop and Define Tool Specific Renovation Strategy

Once a conversion tool is selected, procedures must be written to enumerate the explicit steps required of the renovator. Like other process definitions, this ensures a consistent approach among all renovators. In some cases, the steps are complex, with subsequent steps dependent upon the success of the previous steps.

Specific Actions:

1. **Enumerate the explicit steps required of the renovator to enable use of the selected renovation tool**.
   The objectives of this is to ensure that all renovators use the selected tool consistently and that the Atailoring@ process is clearly defined and documented.

Required IT Deliverable: Renovation Tool Procedures	Responsible: Project teams & Year 2000 Program Office

E.e. Conduct Conversion Pilot, Document Areas for Improvement

Pilot projects are an important element in improving estimates and approaches. Ideally, pilots are conducted during the planning phase, and prior to finalizing the budgets and plans.

Specific Actions:

1. **Identify a pilot candidate**.

2. **Apply the renovation methodology to the candidate software and data**.

3. **Implement the pilot**.

4. **Document issues, and lessons learned**.
   Apply lessons learned to the methodology, where possible.

5. **Judge the impact.**

6. **Address plan and schedule changes.**

Optional Deliverable: Results of Conversion Pilot	**Responsible: Project teams**
Optional Deliverable: Shared Results of Pilot	**Responsible: Year 2000 Program Office**

Objective F. Establish a Year 2000 Environment

Having an environment suitable for executing the conversion tasks and validation of the converted software is of paramount importance.

Strategies:

F.a. Identify Environment Requirements for Hardware and Software

Specific Actions:

1. **Determine the overall mainframe strategy for conversion implementation**. One common strategy is to use existing mainframe capabilities (supplemented with conversion tools) to conduct the actual conversion, and to use an additional mainframe (or LPAR) to conduct validation and forward-date testing. The strategies that are used for testing will define the requirements for an Independent Testing Facility (ITF). The existing mainframe must be improved to contain compliant language environments (compilers, debuggers, etc.), CM tools, and conversion tools. The additional mainframe (or LPAR) could be a site configured much in the same way as the existing production mainframe, with constraints of Acompliant only@ software, and forward-date testing capability.

Required IT Deliverable: Overall Mainframe Strategy (Architecture Diagram)	**Responsible: Systems Operations Division** **Support: Year 2000 Program Office**

2. **Identify all software components for each hardware component.**
   Once the overall mainframe strategy is identified, the precise configuration must be determined. Identify all software and hardware components for each region (DVL, SAT, QAX, TRN, PRD).

Required IT Deliverable: Mainframe Configurations (Lists of Components) for each region.	**Responsible: Systems Operations Division**

3. **Prepare an environment Implementation Plan.**
   The plan should include all components identified in F.a.1 and F.a.2, along with a planned schedule for implementation.

Required IT Deliverable: Year 2000 Environment Implementation Plan	**Responsible: Systems Operations Division** **Support: Year 2000 Program Office**

F.b. Monitor Environment Resources

Specific Actions:

1. **Develop procedures to monitor the environment resources**.

Required IT Deliverable: Procedures to monitor the Environment resources.	**Responsible: Systems Operations Division**

Objective G. Manage Risk

Strategies:

G.a. Identify Risks and Risk Mitigation

Consistent with Treasury guidance on risk management for Year 2000 conversion programs, the Year 2000 Program Office requires each process owner to develop and monitor a Risk Management Plan.
The risk management process consists of six components, which are defined in Table 3.

Specific Actions:

1. **Prepare and maintain a risk management spreadsheet.**
   The spreadsheet should be prepared in accordance to the specifications defined in Table 3.

Required IT and Non-IT Deliverable: Risk Management Spreadsheet	**Responsible: Process Owners** **Support: Project teams, Year 2000 Program Office**

Table 3. Components of Risk Management

Component	Definition
1. Identification	Identify the risk in specific, singular terms. Example: *Vendor product has incompatible compliance definition*. The Andersen Vulnerability Assessment report is a good source of risk items.
2. Characterization	Risk is characterized in terms of type, probability and impact. **Type**: cost, schedule, management or technology. **Probability**: certain, possible, or unlikely. **Impact**: critical, severe, moderate, minor or minimal.
3. Analysis	Risk is a function of uncertainty, and must be analyzed in terms of response options. **Assumption**: if the risk is acceptable, the simplest response is to assume the risk. **Avoidance**: the preferred response for unacceptable risk. **Mitigation**: if the risk is unacceptable, and cannot be avoided, the risk must be managed, or controlled, by mitigation.
4. Mitigation	Based upon risk probability and impact, select an acceptable response approach.
5. Mitigation Threshold	Mitigation occurs when the risk factors exceed a defined threshold.
6. Monitoring	The spreadsheet will be monitored monthly, or as otherwise required, for changes in risk status that could alter the selected response option.

Appendix G presents an embedded template for preparing a risk management spreadsheet.

2. **Monitor the identified risks.**
   As time continues and progress is tracked, risk factors may change. Initiate the mitigation when the defined threshold is reached.

Required IT and Non-IT Deliverable: Updated Risk Management Spreadsheet	**Responsible: Process Owners**

Objective H. Plan for System Continuity

Strategies:

H.a. Prepare System Contingency Plan

Continuity of essential Customs operations through the Year 2000 rollover must be assured. Because the development of contingency plans for all systems is required by GAO, Andersen Consulting, LLP has prepared guidelines to assist Treasury bureaus, like Customs, in the development of contingency plans for their mission-critical systems. The guidance is included in this document, as Appendix H, Treasury Department Year 2000 Contingency Planning. The actions described below are further described in Appendix H.

Specific Actions:

1. **Determine the time horizon-to-failure dates for each mission-critical system or sub-system.**
   This is the earliest point at which a system failure is expected to result from a date-related error. This date could be much earlier, or even somewhat after January 1, 2000. Only if the applications will all be compliant before their respective failure dates will the system be compliant Ain time@; otherwise, it must be assumed that it will fail and that service interruptions can occur.

2. **Determine the estimated compliance date of the mission-critical system or sub-system.**
   Using the detailed renovation plan (D.b), determine the estimated compliance date of the mission-critical system or sub-system This date will drive the decision of when to initiate the secondary systems.

3. **Identify one or more secondary, or fallback options.**
   These are the possible contingencies or alternative approaches that could be used to sustain the business function in the absence of the primary system. These may be other Government systems, commercial software, outsourcing, and manual procedures, in any combination.

4. **Determine the time needed to implement each fallback option.**
   For each fallback option identified in step 3, determine how long it will take to put it into place. This timing will drive the decision of when to begin implementing the fallback option.

5. **For each fallback option, develop continuity scenarios**. Depending on the various dates determined in steps 1, 2 and 4, there are different outcomes regarding fallback action to be taken. Three scenarios, more fully described in Appendix H, address some of the circumstances likely to be encountered:

   a. Renovation Completed Early

b. Renovation Completion at Risk
c. Renovation Completed Late

Required IT and Non-IT Deliverable: System Contingency Plan See Appendix H.	**Responsible: Process Owners** **Support: Project teams, Year 2000 Program Office**

Objective I. Assess and Renovate Systems

The detailed design phase of a Year 2000 conversion project is similar in many respects to that of any development project. This phase finalizes the specifications for conversion, and executes the detailed procedures, using disciplined process controls.

Strategies:

I.a. <u>Identify Interfaces</u>

Internal interfaces are those that communicate with other systems within the Customs portfolio (e.g., program to program). The techniques for identifying internal interfaces may differ from external interfaces, but having a complete inventory remains an essential conversion success factor.

External interfaces communicate with sources outside of Customs such as: Trade, other government agencies, etc. The Year 2000 Program Office must prepare an inventory of external interfaces to include the media used, format, and the external parties that receive the information. It is suggested that an automated tool to identify date issues within the interfaces. The Year 2000 plan will specify the tasks, process, and coordination necessary to maintain the interfaces. Prepare an Interface Renovation Plan, that specifies how each interface definition should be modified.

Specific Actions:

1. **Identify files used externally and internally.**

Required IT Deliverable: Interface Inventory	**Responsible: Project teams** **Support: Process Owners, Year**

3. **Identify actions to be taken.**
   Identify in accordance with the Compliance Approach Definition defined in E.c.2.

4. **Prepare an Interface Renovation Plan**.
   The plan should identify the interface points, define the renovation specification, and the baseline schedule metrics as in D.b.2.

Deliverable: Interface Renovation Plan, to include interface recipients, interface specification, and renovation specification.	**Responsible: Project teams**

I.b. Secure Interface Agreements

Written memoranda between Customs process owners and external partners may be required (e.g., MOUs). Each project team, in combination with the corresponding process owners, must identify the external partners requiring written memoranda regarding the external interfaces. The Year 2000 Program Office will support the project teams and process owners in securing the necessary agreements with external partners.

Specific Actions:

1. **Identify the external partners requiring written memoranda.**
   Use the Interface Inventory developed in I.a.

2. **Submit memoranda requirements to the Year 2000 Program Office**. The Year 2000 Program Office will coordinate with the external partner, and the process owner to secure the necessary interface agreements.

Requests for exemption from standard date formats must be submitted to the Year 2000 Program Office with an accompanying Date Standard Waiver Form. Reference Appendix I for a copy of the form.

I.c. Identify Specific Problematic Areas *(SDLC Counterpart: Detailed Specification)*

Having developed a detailed renovation schedule (D.b.2), and having defined the specific compliance approach for the team (E.c.2), the renovation analysts must evaluate the non-compliant software, data, or non-IT item to specify the exact renovation solution.

Specific Actions:

1. **Specify the exact renovation solution.**
   Using the selected automated tool (E.b), and the procedures for renovation analysis (E.d), specify the exact renovation solution. The Year 2000 Program Office expectation is that the project teams have subdivided the total workload so that one or more renovation analysts, can prepare specifications suitable for one or more renovation programmers.

   For non-IT systems, the specification of solution will generally come in the form of Replace or Retire. For each non-IT item, specify the solution.

Required IT and Non-IT Deliverable: Renovation Specifications	**Responsible: Project teams**

I.d. Renovate the Systems *(Develop and Unit Test)*

Specific Actions:

1. **Renovate the system or asset.**
   Using the selected automated tool (E.b), and method (E.d) renovate the code. For non-IT items, track the vendor name and performance measures of item replacement.

Required IT and Non-IT Deliverable: Renovated Software, Renovated Data, or Replaced Non-IT item	**Responsible: Project teams**

2. **Build the renovated release.**
   Using the language environment tools (compilers, linkers, etc.), build the renovated release.

Required IT Deliverable: Renovated Release	Responsible: Project teams

3. **Prepare a unit test plan**
   Using the detailed renovation specification (I.c.1) as guidance, prepare a unit test plan, exercising the specific changes made.

Required IT Deliverable: Unit Test Plan	Responsible: Project teams

4. **Execute the unit test plan.**
   Compare the actual test results to the expected results. Continue the test activity until the actual results are either equivalent to the expected results, or if different, can be explained.

I.e. <u>Monitor Commercial off the shelf (COTS) Compliance</u>

This objective component applies to several organizational entities, monitoring COTS compliance from different perspectives. Some will be monitoring compliance of products used by the entire organization, while others will monitor compliance of system-specific products. The Year 2000 Program Office will serve as the central information repository for all Year 2000 compliant COTS products.

Specific Actions:

1. **Identify the installed COTS or licensed products**.

2. **Prepare a spreadsheet.**
   The spreadsheet similar to Appendix F makes note of several pertinent attributes of the COTS product.

Required IT and Non-IT Deliverable: Vendor Compliance Record	Responsible: Project teams, Systems Operations Division, Year 2000 Program Office

3. **Identify issues.**
   Identify issues such as late and unacceptable deliveries. The issue may need to be identified as a specific risk on the risk management spreadsheet.

4. **Update the Vendor Compliance Record**.

Monitor each of the products listed on the spreadsheet for updates to their status of compliance.

Required **Deliverable: Updated Vendor Compliance Record**	**Responsible: Project teams,** **Systems Operations Division,** **Year 2000 Program Office**

Objective J. Validate Systems

Validation compares the functionality of an application to a standard, determines that the changes made are effective, and that no errors are introduced in the conversion process. **The Validation Plan specifies the standard that will be used for validation.**

The IT system validation process has two major goals:

1. To ensure that no errors were introduced during the renovation process,
2. To determine that the application, database, or non-IT item works according to the compliance requirements when integrated with its normal surrounding environment.

There are many ways to approach IT system validation. The Year 2000 Program Office recommends the approach, described at a high level in Figure 2. There are several advantages to this approach. One is that the baseline results can easily be compared to the results of tests on the renovated code with simple comparison tools. At this point in the process, the results should be identical. Another is that by reintroducing the renovated coded in production as soon as it is tested, confidence can be gained in the code by the regular software maintenance team. They will have their hands on the code and will become familiar with it. Also, by taking small, phased steps, it is easier to pinpoint the cause of errors that are revealed by further testing.

The overall approach for non-IT system validation will be detailed in their Validation Plan.

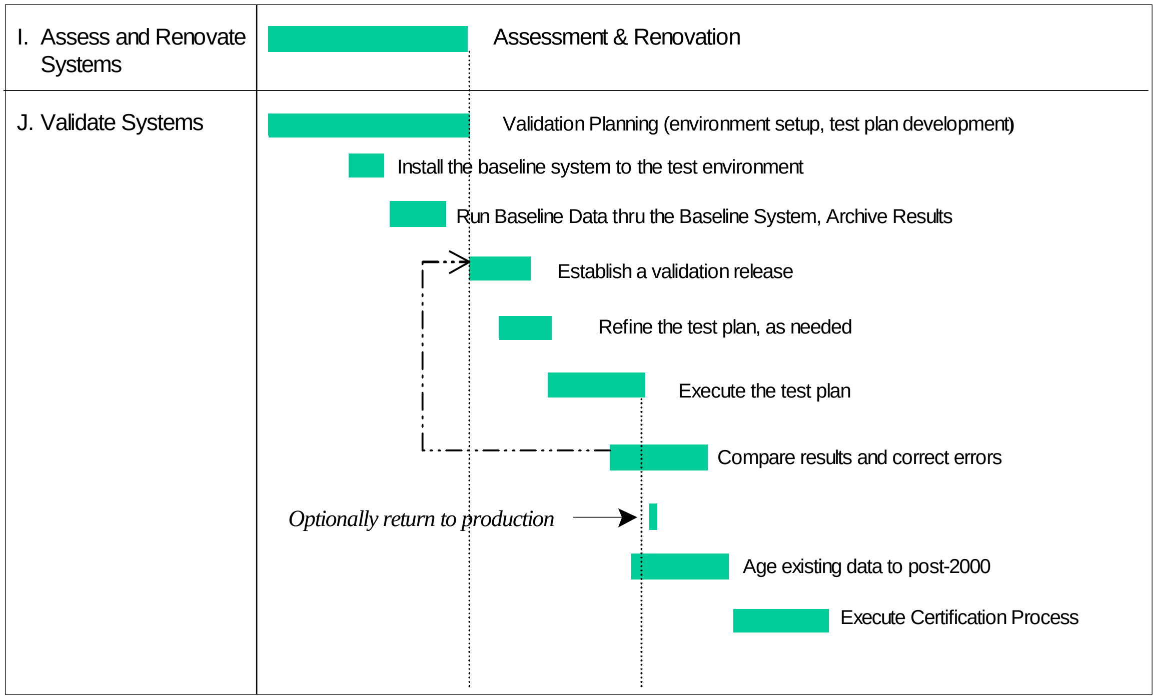
Strategies:

J.a. Validation Planning

The most important message of the Validation Strategy chart (Figure 2) is to convey concurrent activity. Do not wait until the renovation phase is complete before planning for and implementing the validation environment.

Specific Actions:

1. **Identify the validation staff, staff roles and responsibilities, staff relationships**.



J.b.2. <u>Estimate and Acquire Necessary Validation Resources</u>

Given the known renovation workload requirements revealed by the prioritized renovation schedule (D.b.2), estimate validation resource shortfalls, and develop a plan to acquire additional help. The plan must consider the scope of work, and unavoidable lead time (recruiting, security clearances).

Specific Actions:

1. **Estimate the validation team resources**.
   Using the results of the prioritized renovation schedule developed in D.b.2, estimate the project team resources

2. **Develop a plan to acquire additional resources**.
   The Year 2000 Program Office will help with identifying available sources of resources. The plan should identify required resource type, required resource quantity, and the planned acquisition date. Include resources in a pending status. This plan must be updated periodically.

Required IT and Non-IT Deliverable: Validation Resource Acquisition Plan	**Responsible: Validation Teams**

3. **Establish the scope of testing tailored from the Validation Strategy**.
   Testing will invariably uncover errors that are pre-existing errors, which may need different handling than new errors caused by the renovation. Some errors may be acceptable, and have work-around procedures.

4. **Define the backout or recovery procedures.**
   The validation coordinator must be capable of recovering the test environment to the pre-test condition.

5. **Establish the test beds.**
   Knowing the requirements for the code segment being validated, arrange for acceptable data to be transferred to the test environment. This may involve database loads and unloads, or newly developed programs to affect a data migration.

Required IT Deliverable: Validation Test Beds	**Responsible: Validation Teams** **Support: Systems Operations Division**

6. **State all validation objectives and expected outcomes**.
   Use the Compliance Approach Definition (E.c.2) as a basis for the validation objectives.

Required IT and Non-IT Deliverable: Validation Plan	**Responsible: Validation Teams** **Support: Project teams, Year 2000 Program Office**

J.c. <u>Baseline Test</u>

Specific Actions:

1. **Install the baseline system.**
   Install the baseline system (software and data, or non-IT item) to the test environment. This step assumes that there is a logical equivalency, or a 1:1 correspondence between the baseline system and the renovated system needing validation. This step also assumes that proper configuration management procedures have been followed.

2. **Run the baseline data thru the baseline system**.

3. **Archive the baseline results**.

Required IT Deliverable: Baseline Test Results Report	**Responsible: Validation Teams** **Support: Project teams**

J.c. Validation Test

Specific Actions:

1. **Establish the validation release**.
This action results in a fully executable system. The project team has completed successful unit tests for each of the system components, and has executed the renovated (if necessary) build procedures. The release and all of its components must be identifiable by means of a release number, as specified in the CM Plan.

Required IT Deliverable: Validation Release	**Responsible:Project teams** **Support: Validation Teams**

2. **Refine the validation plan.**
Before executing the validation plan, make any necessary adjustments to accommodate excepted conditions from prior executions.

3. **Execute the validation plan.**
The validation plan is executed against the renovated code using the baseline test data.

Required IT and Non-IT Deliverable: Validation Test Results Report	**Responsible:Validation Teams** **Support: Project teams**

4. **Compare the baseline results to the validation test results**.
The results should be identical.

B Resolve all differences.

B Fix any errors that are uncovered.

B Make any necessary refinements to the validation plan.

B Re-test.

Objective K. Verify Systems

The verification step is the final opportunity to review applications before they are re-introduced into the production environment. **Final compliance verification requires that the renovated systems be tested with post-Year 2000 date values**. Systems cannot be verified Year 2000 compliant unless data that reflects events occurring after the year 2000 have been successfully processed through the systems. The Year 2000 Program Office recommends a pro-active approach to verification, which means that existing data must be forward aged, so that the dates reflect values beyond 1999. This step assumes that all software, whether application systems or operating systems or ancillary support software, is either known, or believed to be Year 2000 compliant. This environment is analogous to a Aclean room@.

Strategies:

K.a Forward Age the Validation Data

Specific Actions:

1. **Modify the data used for validation to reflect dates beyond the system failure date**.
The forward aged data will be used to verify the validated release for use in the next century.

K.b Verification

Specific Actions:

1. **Define the verification criteria**.
Use the Compliance Approach Definition (E.c.2) as a basis for the criteria.

2. **Install the renovated and validated release to the Year 2000 environment.**

3. **Test the validated release with the forward aged data**.

Required IT and Non-IT Deliverable: Verification Results Report	**Responsible: Validation Teams** **Support: Project teams**

Objective L. Implement Systems to Production

The fully verified system or application must be re-introduced into the production environment. During this phase, the current production data is transformed according to data expansion requirements, the verified software is moved into production, and the system users are made aware of the cutover date.

Strategies:

L.a Prepare an Implementation Plan *(SDLC Counterpart: Implementation)*

While most implementation plans include elements for hardware and system software upgrades, what is needed here is an integrated schedule of events that allows for an orderly transition of system objects from the test environment into the production environment.

Specific Actions:

1. **Conduct coordination meetings with the affected project teams.** Identify the dependencies, and schedule the actual implementations in the order of least dependent to most dependent.

Required IT and Non-IT Deliverable: **Implementation Plan**	**Responsible:**	**Systems Operations Division**
	Support:	**Validation Teams** **Project teams**

L.b Prepare a Training Plan

Operational procedures are taught to the users and maintainers of the renovated systems.

Specific Actions:

1. **Identify staff members in need of training.**
   Schedule classes, prepare training materials, train the trainers, reserve classroom space and computers.

2. **Revise software, documentation or procedures.**
   Training may reveal errors in the software, documentation, or procedures. Report all errors via the change control process.

Required IT and Non-IT Deliverable: Training Plan	**Responsible:**	**User Support Division**
	Support:	**Validation Teams** **Project teams**

L.c Convert data, as required

Where date fields were expanded, the data must be converted and moved. This guidance assumes that proper validation is complete on the process used to convert and move the data. **This is the final data conversion execution**.

Specific Actions:

1. **Freeze access to the current production data.**
2. **Execute the data extraction or unload process from the current production environment.**

3. **Implement the database definition, using the renovated definitions.**

4. **Execute the data posting, or load process into the new production environment.**

5. **Restore access to the new production data.**
   Ensure that all current users and bind plans have appropriate privileges to the new environment.

L.d Move verified systems to the production platform

This part of the process brings together the validated software with data in the new version of the production database.

Specific Actions:

1. **Install the renovated and validated software to production.**
   Using the configuration management procedures, install the renovated and validated executable software to the production environment.

L.e Cutover to production

All renovation work is now complete. The formal cutover date is determined and announced.

3.0 Operational Checklist

Objective A. Centrally Manage The Year 2000 Program

1. Establish a Year 2000 Program Office
2. Establish an Executive Council
3. Establish and Track the Overall Program Plan
4. Develop Reporting Requirements
5. Year 2000 Performance Measures Guideline

Objective B. Ensure Year 2000 Awareness

1. Outreach Plan
2. Implement the Outreach Plan
3. Maintain a Central Database of Program Artifacts

Objective C. Establish the Enterprise Mission-Critical Inventory

1. Inventory Methodology for Software, Data, and Non-IT items
2. Baselined Software Inventory
3. Baselined Data Inventory
4. Baselined Non-IT Inventory
5. Updated Inventories, based upon results of cross-reference analysis

Objective D. Plan for Conversion Resources

1. Mission-Critical Component List
2. Prioritized Renovation Schedule
3. Migration Plan
4. Resource Acquisition Plan

Objective E. Determine Conversion Methodology

1. Configuration Management
2. Quality Assurance Procedures
3. Tool Recommendations
4. Compliance Approach Definition
5. Identify Necessary Bridges
6. Conversion Tool Procedures
7. Results of Conversion Pilot (Optional)
8. Shared Results of Pilot (Optional)

Objective F. Establish a Year 2000 Environment

1. Overall Mainframe Strategy (Architecture Diagram)
2. Mainframe Configurations (Lists of Components)
3. Year 2000 Environment Implementation Plan

Objective G. Manage Risk

1. Risk Management Spreadsheet
2. Updated Risk Management Spreadsheet

Objective H. Plan for System Continuity

1. System Contingency Plan

Objective I. Assess and Renovate Systems

1. Interface Inventory
2. Interface Renovation Plan
3. Renovation Specifications
4. Renovated Software, Renovated Data, or Renovated Non-IT item
5. Renovated Release
6. Unit Test Plan
7. Vendor Compliance Record

Objective J. Validate Systems

1. Validation Resource Acquisition Plan
2. Validation Test Beds
3. Validation Plan
4. Baseline Test Results Report
5. Validation Release
6. Validation Test Results Report

Objective K. Verify Systems

1. Verification Criteria
2. Verification Test Results Report

Objective L. Implement Systems to Production

1. Implementation Plan
2. Training Plan
3. Converted, and Migrated Data
4. Install System to Production Environment
5. Cutover to Production

Appendix A. Summarized IT Inventory

Appendix B. Integrated Mission Critical Schedule

Objective A. Centrally Manage The Year 2000 Program	**Program Office**		**Other Participant (None)**		**ACS**		**TECS**	
	Start	Finish	Start	Finish	Start	Finish	Start	Finish
1. Establish a Year 2000 Program Office	04/1997	05/1997						
2. Establish an Executive Council	05/1997	08/1997						
3. Establish and Track the Overall Program Plan	07/1997	On-Going						
4. Develop Reporting Requirements		10/1997						
5. Year 2000 Performance Measures Guideline		11/1997						
Objective B. Ensure Year 2000 Awareness	**Program Office**		**Other Participant (None)**		**ACS**		**TECS**	
	Start	Finish	Start	Finish	Start	Finish	Start	Finish
1. Outreach Plan		07/1997						
2. Implement the Outreach Plan		07/1997						
3. Central Database of Program Artifacts		05/1997						
Objective C. Establish the Enterprise Mission-Critical Inventory	**Program Office**		**Other Participant (None)**		**ACS**		**TECS**	
	Start	Finish	Start	Finish	Start	Finish	Start	Finish
1. Inventory Methodology for Software, Data, and Non-IT items					04/1996	01/1997	04/1996	10/1996
2. Baselined Software Inventory					09/1996	08/1997	09/1996	11/1996

3. Baselined Data Inventory					11/1996	11/1996	11/1996	11/1996
4. Baselined Non-IT Inventory								
5. Updated Inventories, based upon results of cross-reference analysis						On-going		On-going
Objective D. Plan for Conversion Resources	**Program Office**		**Other Participant (None)**		**ACS**		**TECS**	
	Start	Finish	Start	Finish	Start	Finish	Start	Finish
1. Mission-Critical Component List					05/1997			
2. Prioritized Renovation Schedule					08/1997	08/1997	08/1997	08/1997
3. Migration Plan	09/1997	10/1997			09/1997	10/1997	09/1997	10/1997
4. Resource Acquisition Plan					08/1997	08/1997		
Objective E. Determine Conversion Methodology	**Program Office**		**Other Participant (QAT)**		**ACS**		**TECS**	
	Start	Finish	Start	Finish	Start	Finish	Start	Finish
1. Configuration Management Procedures	10/1997	10/1997	08/1997	10/1997				
2. Quality Assurance Procedures	10/1997	10/1997	06/1997	10/1997				
3. Tool Recommendations		On-going		On-going		On-going		On-going
4. Compliance Approach Definition	09/1997	10/1997			09/1997	10/1997	09/1997	10/1997
5. Identify Necessary Bridges					11/1997		11/1997	
6. Conversion Tool Procedures					02/1997	On-going	02/1997	On-going
7. Results of Conversion Pilot (Optional)								
8. Shared Results of Pilot (Optional)		08/1997						
Objective F. Establish a Year 2000 Environment	**Program Office**		**Other Participant (Systems Operations Divsion)**		**ACS**		**TECS**	

	Start	Finish	Start	Finish	Start	Finish	Start	Finish
1. Overall Mainframe Strategy (Architecture Diagram)								
2. Mainframe Configurations (Lists of Components)								
3. Year 2000 Environment Implementation Plan								
Objective G. Manage Risk	**Program Office**		**Other Participant (None)**		**ACS**		**TECS**	
	Start	Finish	Start	Finish	Start	Finish	Start	Finish
1. Risk Management Spreadsheet								
2. Updated Risk Management Spreadsheet								
Objective H. Plan for System Continuity	**Program Office**		**Other Participant (Process Owners)**		**ACS**		**TECS**	
	Start	Finish	Start	Finish	Start	Finish	Start	Finish
1. System Contingency Plan					10/1997	12/1997	10/1997	12/1997
Objective I. Assess and Renovate Systems	**Program Office**		**Other Participant (None)**		**ACS**		**TECS**	
	Start	Finish	Start	Finish	Start	Finish	Start	Finish
1. Interface Inventory	03/1997							
2. Interface Renovation Plan								
3. Renovation Specifications					08/1997	On-going	08/1997	On-going
4. Renovated Software, Renovated Data, or Renovated Non-IT item					01/1997	10/1998	01/1997	10/1998

5. Renovated Release (Ready for Validation)			▨	▨				
6. Unit Test Plan			▨	▨				
7. Vendor Compliance Record								
Objective J. Validate Systems	**Program Office**		**Other Participant (Quality Assurance Teams)**		**ACS**		**TECS**	
	Start	Finish	Start	Finish	Start	Finish	Start	Finish
1. Validation Resource Acquisition Plan								
2. Validation Test Beds								
3. Validation Plan								
4. Baseline Test Results Report								
5. Validation Release								
6. Validation Test Results Report								
Objective K. Verify Systems	**Program Office**		**Other Participant (Quality Assurance Teams)**		**ACS**		**TECS**	
	Start	Finish	Start	Finish	Start	Finish	Start	Finish
1. Verification Criteria								
2. Verification Test Results Report								
Objective L. Implement Systems to Production	**Program Office**		**Other Participant (Quality Assurance Teams)**		**ACS**		**TECS**	
	Start	Finish	Start	Finish	Start	Finish	Start	Finish
1. Implementation Plan								
2. Training Plan								

3. Converted, and Migrated Data								
4. Install Systems to Production Environment								
5. Cutover to Production								

Appendix C. Year 2000 Program Office Reporting Requirements

To achieve the Departmental goals, the Customs Year 2000 Program Office must establish performance indicators, then measure progress against the indicators. Performance indicators are categorized as scope, schedule, and resources. The Customs Year 2000 Program Office will collect metrics to support performance measurement, as follows:

Scope The Year 2000 Program Office will track the number of units:

Software Units:

$ total quantity of programs
$ quantity of programs in progress
$ quantity of programs complete[4]

$ total software lines of code (SLOC)
$ quantity SLOC in progress
$ quantity SLOC complete

Data Units:

$ total quantity of data tables[5]
$ quantity of data tables in progress
$ quantity of data tables complete
$ total quantity of table occurrences
$ quantity of table occurrences migrated

Non-IT Units:

$ total quantity of non-IT units
$ quantity of non-IT units in progress
$ quantity of non-IT units complete

Schedule The Year 2000 Program Office will track schedule variance:

[4] For purposes of measurement, a unit is considered complete when it moves from one stage (Assessment, Renovation, Validation, Implementation) to another. Since a unit may go from Assessment to Implementation without going to Renovation and Validation, the numbers may not track consistently. However, the total should always equal the sum of the parts.

[5] To avoid counting tables (and effort) multiple times, the Project teams should only report the number of tables both used and owned.

Start Date

$ Baseline Start Date — This date is established at the onset of planning and never changes.
$ Planned Start Date — This date is equal to the baseline date until a change is required. This date can change as often as necessary.
$ Actual Start Date — This date is established at the onset of work.

Finish Date

$ Baseline Finish Date — This date is established at the onset of planning and never changes.
$ Planned Finish Date — This date is equal to the baseline date until a change is required. This date can change as often as necessary.
$ Actual Finish Date — This date is established at the completion of work.

Resources The Year 2000 Program Office will track resource usage:

Hours[6]

$ Staff Hours
$ Management Hours

Dollars[7]

$ Staff Dollars
$ Management Dollars
$ Materials

[6] These are estimates of actual labor based on planning algorithms, not actual figures. The Year 2000 Program Office will help the Project teams to formulate consistent algorithms.

[7] These are dollar estimates of actual costs based on planning algorithms, not actual figures. Use an average fully-loaded $/hr for the various labor categories. The Year 2000 Program Office will help the Project teams to formulate consistent algorithms.

SOFTWARE QUALITY ASSURANCE (SQA) PLAN OUTLINE

INTRODUCTION

The purpose of Software Quality Assurance (SQA) is to provide Year 2000 Program Management with appropriate visibility into the process being used by Customs mission-critical software projects and of the products being built for the Year 2000 Program.

SQA involves reviewing, auditing and monitoring the software products and activities to verify that they comply with the applicable procedures and standards by providing the Year 2000 Program Office (PO) and other appropriate managers with the results of these reviews, audits and monitoring activities.

The Year 2000 SQA Team works with the Year 2000 PO during its early stages to establish plans, standards, and procedures that will add value to the PO and satisfy the constraints of the Year 2000 Program and its organization=s policies. By participating in establishing the plans, standards, and procedures, the SQA team helps ensure they fit the Year 2000 Program needs and verifies that they will be usable for performing reviews and audits throughout Customs mission-critical software projects. The SQA Team reviews project activities and audits software work products throughout the Year 2000 Program process and provides management with visibility as to whether Customs software projects are adhering to their established plans, standards, and procedures.

Compliance issues are first addressed within the Customs mission-critical software projects, themselves, and resolved there if possible. For issues not resolvable within the Year 2000 Program and/or the Customs mission-critical software projects, the SQA Team escalates the issue to an appropriate level of management for resolution.

SQA PLAN

An SQA Plan is prepared for the Year 2000 Program Office according to a documented procedure. This procedure typically specifies that: the SQA plan is developed in the early stages of, and in parallel with, the Year 2000 overall project planning. The SQA plan is reviewed by the affected groups and individuals. The SQA plan is managed and controlled. AManaged and controlled@ implies that the version of the product in use at a given time (past or present) is known (i.e., version control), and changes are incorporated in a controlled manner (i.e., change control). If a greater degree of control than is implied by Amanaged and controlled@ is desired, the work product can be placed under the full discipline of configuration management, as described in the Software Configuration Management Plan for Year 2000.

The SQA Team=s activities are performed in accordance with the SQA Plan. The SQA Plan is created after the SQA Team participates in the preparation and review of the Year 2000 management plans, standards and procedures processes.

The plan covers:

1. Responsibilities and authority of the SQA Team.

2. Resource requirements for the SQA Team (including staff, tools, and facilities).

3. Schedule and funding of the Year 2000 SQA Team=s activities.

4. The SQA Team=s participation in establishing management plans, standards, and procedures for the Year 2000 Program.

5. Evaluations to be performed the SQA Team. Examples of products and activities to be evaluated include: mission-critical operational software and support software; deliverable and nondeliverable products; software and nonsoftware products (e.g., documentation); product testing and product validation activities (e.g., executing test cases); and the activities followed in creating the renovated product.

6. Audits and reviews to be conducted by the SQA group.

7. Project standards and procedures to be used as the basis for the SQA Team=s reviews and audits.

8. Procedures for documenting and tracking noncompliance issues to closure. These procedures may be included as part of the plan or may be included via reference to other documents where they are contained.

9. Documentation that the SQA Team is required to produce.

10. Method and frequency of providing feedback to the Year 2000 Program Office and Customs mission-critical software projects and other software-related groups on SQA activities.

SOFTWARE CONFIGURATION MANAGEMENT (CM) PLAN OUTLINE

INTRODUCTION

The purpose of Software Configuration Management (SCM) is to establish and maintain the integrity of the products of the Year 2000 Program Office (PO) and Customs mission-critical software projects throughout the five phased process.

SCM involves identifying the configuration of the software (i.e., selected software work products and their descriptions) at given points in time, systematically controlling changes to the configuration, and maintaining the integrity and traceability of the configuration throughout its five-phased process. The work products placed under SCM include the software products produced under the renovation phase and validation phase and the items that are identified with or required to create these software products (e.g., the compiler).

A software baseline library is established containing the software baselines as they are developed. Changes to baselines and the release of software products built from the software baseline library are systematically controlled via the change control and configuration auditing functions of SCM.

SCM PLAN

A SCM Plan is prepared for the Year 2000 Program Office according to a documented procedure. This procedure typically specifies that the SCM Plan is developed in early stages of, and in parallel with, the Year 2000 overall program planning. The SCM plan is reviewed by the affected groups and is managed and controlled. A documented and approved SCM plan is used as the basis for the performance of SCM activities.

The SCM Plan covers the SCM activities to be performed, the schedule of activities, the assigned responsibilities, and the resources required (including staff, tools, and computer facilities). The SCM Plan also covers the SCM requirements and activities to be performed by the Year 2000 PO and by Customs mission-critical projects.

The SCM Plan will provide the requirements for the establishment of a CM library system which will act as a repository for all the Year 2000 baselines; requirements for the change control process; the requirements for storage and retrieval of configuration items/units; the requirements for sharing and transfer of configuration items/units between the affected groups and between control levels within the library; the requirements for storage and recovery of archive versions of configuration items/units; requirements for the creation, storage, update, and retrieval of SCM records; requirements for status accounting; requirements for audit activities; and requirements for the maintenance of the library structure and contents. Examples of library maintenance functions include: backup/restoring of library files; and recovery from library errors.

Renovation Approach Describes the method used to achieve compliance, and how each type of field and operation should be handled. It also defines acceptable exceptions to the rules.	Interpret the century year using pivot year, rather than expanding fields. This is a procedural approach. C Use 45 as the pivot year (00-45 is 2000-2045, 46-99 is 1946-1999). C Leave all internal fields in 2-digit format except birth-year fields: C Retain as 2-digit data C Interpret using pivot year for calculations C Expand for display on reports and panels C Update all code logic performing date operations. C Expand date fields used in sorting from 6 digits to 8 digits.
Interface Approach Describes how date data in shared files will be handled	C Do not expand any data unless used as an index or within a sort key. C Accept expanded dates when required by other applications, but continue to use 2-digit year formats internally. This is a bridge approach.
Data Approach Describes how the dates are to be formatted on screens and reports	C CICS panels: Leave all years in 2-digit format except birth years, which should be expanded to 4-digits, using the pivot year to calculate. C Reports: Leave all years in 2-digit format except birth years, which should be expanded to 4-digits, using the pivot year to calculate.

Appendix G. Risk Management Spreadsheet Template

Identified Risk	Risk Characterization			Response	Mitigation Threshold	Risk Mitigation
	Type	Probability	Impact			

Appendix H. Treasury Department Year 2000 Contingency Planning

At a year 2000 conference organized by the Software Productivity Group, Mr. William Ulrich, president of consultancy Tactical Strategy Group, stated the following; A*Every mission-critical function that relies on a computer system should have a contingency plan.*@

The purpose of this document is to provide general guidance for Treasury Bureaus and Offices when preparing Year 2000 Contingency Plans. It is recognized that the contingency planning process already may be underway; therefore, this guidance will not be prescriptive in nature, but rather will address issues that should be considered during this process. A sample Contingency Plan is also provided to demonstrate the key issues that should be addressed and included in each Bureau and Office Plan.

Background

Section 6.4 of the Treasury Year 2000 Date Conversion Program Management Plan (December 18, 1996) discusses Contingency Plans under AReports and Products@. The contents will not be repeated in this document, but are available in the Treasury Plan.

Contingency Plans should be developed for all Treasury systems; however, the emphasis must be on mission-critical systems. Therefore, one of the initial considerations is the question of mission-criticality. This determination should be based on an inventory of systems and applications.

There are two general types of Contingency Plans: Disaster Recovery Plans, and Continuity of Operations (COOP) Plans. As applied to year 2000 projects, these two plans are interpreted as follows: a Year 2000 Disaster Recovery Plan applies to unanticipated failure of mission-critical systems that were believed to be year 2000 compliant; a Year 2000 COOP Plan applies to the sustainment of service where mission-critical systems may not be compliant in time. Each type of plan will be discussed in more detail later in this document.

Time Horizon-to-Failure

The distinction between types of Year 2000 Contingency Plans given above relies on achieving compliance in time to avoid a system failure. What constitutes Ain time@ requires some additional clarification before continuing.

There are a number of milestones that should be considered. These include critical date changes, of course, such as the start of Federal fiscal 2000 (October 1, 1999); the start of calendar 2000 (January 1, 2000); and the start of many state fiscal 2000s (July 1, 1999). There are also date- associated milestones established by Treasury, OMB, and other agencies. Unfortunately, there is no one specific fixed date that can be guaranteed to be Ain time@ to avoid a failure. Instead, one must consider the concept of Atime horizon-to-failure.@ Simply put, this is the earliest point at which a system failure is expected to result from a date-related error. This expected system failure date could be much earlier, or even somewhat after January 1, 2000.

For example, the Social Security Administration experienced their first time horizon-to-failure in the fall of 1989 (fiscal 1990) when performing 10-year payout forecasts (through fiscal 2000). On the other hand, some IRS systems may not experience date failures until tax year 2000 returns begin to be filed. The point here is that, for every mission-critical system, all applications must be analyzed to determine the predicted failure date. Only if these applications will all be compliant before their respective failure dates will the system be compliant Ain time@; otherwise, it must be assumed that it will fail and that service interruptions can occur.

Disaster Recovery Plans

Treasury Bureaus and Offices should already have Disaster Recovery Plans in place for their mission-critical systems that address catastrophic losses through fire, flood, sabotage, and similar service interruptions. There are unique characteristics of year 2000 failures, however, that must be addressed by any Disaster Recovery Plan. Disaster recovery site computers typically execute the same software as that which caused a failure in the operational system. Recovery from a date-error failure, therefore, will be more problematic than just restoring operations at a remote site.

The first problem is that of error detection. While many processing errors may cause abnormal program termination or generate operator error messages, other errors will simply yield incorrect results. Therefore, vigilance will be necessary at each critical system date. This includes round-the-clock technical support, increased quality checks, and sampling and verification of results.

Once a Ayear 2000" problem is detected, the technical staff must isolate the fault, correct the error, and test the modified application prior to putting it back into production. This will likely be accomplished according to established maintenance procedures. Since it is probable that more than one error will occur and be found during date-critical periods, an emergency response team should be readily available.

An important aspect of service restoration will be the question of data integrity. It is possible that the original fault may corrupt date fields in files and data bases. A regular schedule of data backup will be especially important during critical system dates. If a disaster recovery site is part of the overall backup process, additional care must be taken to avoid contaminating that site=s data with corrupted production system data. In the event of a catastrophic failure, the recovery site may be the source for data essential to recovery of the production system, so it is important to carefully manage this potentially critical resource.

Continuity of Operation (COOP) Plans

The Treasury Year 2000 Program Management Plan discusses options for suspension or termination of service when systems will not be year 2000 compliant before their anticipated failure date. Service must be sustained for mission-critical systems, however, although some degradation may be acceptable.

The purpose of a COOP Plan is to establish roles, responsibilities, service levels, and support systems to be employed if the primary automated system is not operational. A COOP Plan identifies decisions that must be made on a date-driven basis. The first of these dates is the time horizon-to-failure (i.e., expected failure date), discussed previously in this document.

Just as important as the expected failure date is the estimated compliance date for the mission-critical system; this is the actual date by when the system conversion or replacement is planned to be complete. It should be noted that, since this date will drive the decision of when to initiate alternative sustainment measures, it should be as accurate as possible. It should be based on the size of the system, progress to date, availability of other required systems or components, etc.

The third, go/no-go date is based on time needed to institute secondary systems or alternative sustainment measures. Determining this date requires that one or more alternatives have been identified that could be employed to sustain critical functions in the absence of the primary system. These may include other Government systems, commercial software, outsourcing, and manual back-up procedures, in any combination. Each sustainment alternative should be evaluated in terms of level of service offered, cost, and lead time to implement. Good strategy may be to consider at least one full-service option (probably expensive and long-lead), and one limited-service option (lower cost and quicker).

For each system fall-back option considered, the go/no-go date is the latest point in time that the alternative can be initiated to achieve sustainment before the anticipated primary system failure. In other words, the fall-back plan must be executed by the no/no-go date to ensure that there will be no service interruption in the event that the primary system cannot be made year 2000 compliant before the expected failure date. The choice of the specific option(s) will involve a cost/benefit assessment of the level of service provided, the cost of achieving it, and the expected duration of the sustainment effort. The COOP Plan should clearly establish the decision points and criteria for invoking each option.

Sustainment Scenarios

Depending on the various dates discussed above, there are different outcomes regarding the sustainment action to be taken. Three scenarios are described below that address some of the circumstances that are likely to be encountered.

a. *Conversion Completed Early*. In this scenario the primary system compliance date is expected to be well in advance of the failure date, so the fall-back plan will probably not be needed. The COOP Plan might identify a limited option that could be instituted quickly, but that only needs to meet the minimum required service level for a short time. The primary system is planned to be back in production before the back-up go/no-go point is reached; therefore, the sustainment option would only be initiated in the event of a dramatic slip in the primary system compliance schedule. This is a best-case scenario, as no expenditures are required to prepare for an unnecessary option as long as primary system compliance is achieved before the go/no-go date.

b. *Conversion Completion Risk*. In this scenario the primary system compliance date is expected to be very close to the expected failure date. Thus, it will probably be necessary to initiate the fall-back plan before it is known whether the primary system will achieve compliance in time or not. Here the COOP Plan might identify a moderate service fall-back option to reduce service risk from primary system schedule delays; presumably the associated go/no-go date will occur well before the primary system is due to be back in production. Although this sustainment option may not actually be used, it will be necessary to initiate the fall-back plan as soon as the go/no-go date is reached. In this event, resources might be expended on an option that would never be needed; failure to initiate the fall-back preparations it time, however, could result in loss of critical services if there is any slip in the primary system compliance schedule.

c. *Conversion Completed Late*. In this scenario the primary system compliance date will definitely be after the expected failure date. Thus, the COOP Plan must establish a fall-back option that will sustain an adequate level of service until the primary system is compliant and available again. In this instance, the go/no-go date is basically the latest point at which the fall-back option can be initiated in order to avoid service interruption. This is a worst-case scenario since it essentially requires two systems (primary and back-up) that must be assured to be year 2000 compliant. Some long-term benefit may be gained, however, by incorporating the fall-back sustainment procedures into the Disaster Recovery Plan for the primary system.

Evaluating Contingency Plans

The preceding discussion of Disaster Recovery Plans and COOP Plans was intended to provide general guidance. In addition, a set of questions is listed below that identify specific issues that should be addressed by Bureaus and Offices in their own Contingency Plans.

1. Do all mission-critical systems have Contingency Plans?

2. Has the Atime horizon-to-failure@ been established for each mission-critical system?

3. Is the compliance schedule status being tracked on a regular basis (e.g., monthly) for each mission-critical system?

4. Have service sustainment options been identified for each mission-critical system?

5. Has a go/no-date been established for each service sustainment option?

6. Do the Contingency Plans clearly establish the roles, responsibilities, criteria, decisions, resources, and actions required to initiate each service sustainment fall-back option?

7. Do the go/no-go date decisions provide an adequate Abuffer zone@ before the expected failure date of each mission-critical system to allow for unforseen circumstances?

8. Have the fall-back options previously been used and verified? If not, does the schedule provide adequate time for their verification, and modification if necessary?

9. Have inter-system and supply chain dependencies been identified for each mission-critical system? Have their year 2000 compliance status and schedule been determined?

10. Have alternate mechanisms or sources of critical information or supplies been established?

11. Has a technical Aemergency response@ team been established for each mission-critical system?

12. Have error detection and recovery procedures been augmented to be used during the critical dates of each mission-critical system?

13. Have additional data backup procedures been established to be used during the critical dates of each mission-critical system?

14. Does the Contingency Plan include a role for the disaster-recovery site (e.g., sequestered back up data)?

15. Have precautionary measures been established to preclude contamination of the disaster-recovery site data in the event of mission-critical system errors?

Contingency Planning Example

The following example is provided for a hypothetical system. While it doesn=t address general issues of organization or roles and responsibilities, it does demonstrate some of the key issues related to sustainment alternatives and criteria. Each Bureau and Office should include similar information for each of their critical systems, in addition to their unique organizational details.

1. **System Identification**

a. *System Name*. Automated Payroll System (APS)

b. *System Purpose*. At the end of each pay period, time sheets are completed by employees. They are sent electronically to our Automated Payroll System (APS). Employees are than paid using direct deposit to their bank of choice.

c. *System Dependencies*. Every two weeks, income tax deduction data for each employee is sent to the IRS and to respective state tax administrations. Social Security and Medicare deductions are reported to the Social Security Administration. Additionally, direct deposit is used to pay 99% of the employees whose earnings are processed with APS. Electronic time sheets are provided by the internal Electronic Timecard System (ETS).

2. **Year 2000 Risk Assessment**

a. *Estimated Failure Date*. Our failure date analysis initiative determined that the first APS failure associated with the Year 2000 will occur on October 1, 1999, due to the start of fiscal Year 2000.

b. *Impact of Failure*. A failure of APS would result in interruptions to payment of employee salaries, and would jeopardies the accuracy of their federal, state, and local tax liabilities.

c. *System Compliance Status*. A thorough assessment of the APS has been conducted. The conversion is scheduled to begin in January 1998 and last approximately 3 months. In order to focus energy and resources, the Bureau has decided to complete conversions of all mission critical system before testing of any systems. Testing of APS will commence in July 1998 along with all our other systems. The system will be Year 2000 compliant by October 1, 1998, the target date established by the Treasury Department, and one full year before the expected system failure date due to the year 2000.

d. *Dependency Compliance Status*. ETS (which provides inputs to APS) will be converted in 1997; the interface between ETS and APS currently uses four-digit years, and therefore there is no conversion schedule issue. The output from APS will continue to support the current state and federal interface standards, via bridges, until these interfaces adopt the new date standard; after the new standard is adopted, the bridges will be removed.

3. **Service Sustainment Options**

a. *COTS Solution*. There are two products on the market that could suit our needs if APS is not converted and tested in time. One is a product from ABC Corporation, and the other is a product from XYZ Incorporated.

Cost. ABC Product costs $400,000 and XYZ Product costs $250,000. Both solutions are client/server based and can run on existing Year 2000-compliant platforms. However, additional storage would be required at an approximate cost of $20,000.

Schedule. From time of purchase to the date when the COTS solution is fully operational, the transition would take 4 months.

Dependencies. No other exterior organizations should have complications, assuming the transition to this COTS solution achieves the same functionality as the current system.

b. *GOTS Solution*. None of the available GOTS systems appear to satisfy our needs. This is not considered a viable fall-back option.

c. *Manual Solution*. Since the APS was a replacement for a manual process in 1991, it is possible to revert to the previously used process if it were necessary.

Cost. It would be necessary to hire twenty additional staff members in our accounting division until a permanent solution is reached.

Schedule. The transition between the current automated system and this contingency option would take at least six months to implement.

Dependencies. Significant complications could occur in this area. Payroll deduction data must be sent to the IRS, state tax administrations, and Social Security Administration on a biweekly basis; this would require a work around. Hard copies of all time sheets would be required. Hard copy checks would be printed and issued to all individuals.

d. *Alternate Payroll System*. The SSA=s payroll system could be used in the case of a short-term unavailability of APS due to the Year 2000.

Cost. The existing ETS could be adapted to develop a file that can be transmitted to SSA for their use in generating our payroll. This would require an estimated $15,000 in labor.

Schedule. It is estimated that it would only require one month for the bureau to be ready to use the SSA system.

Dependencies. No other exterior organizations would have complications if this solution were used since the SSA payroll system has direct deposit as part of its functionality and also interfaces with the IRS and state tax administrations.

4. **Go/No-Go Decision Date**

The contingency plan must be implemented in April 1998 if the conversion of APS is not yet completed. This is 6 months before the Treasury Department target of October 1, 1998, and would allow any of the options to be exercised so that the payroll function would be back in production by that date.

Appendix I. Date Standard Waiver Form

U.S. CUSTOMS SERVICE
Office of Information and Technology

REQUEST FOR YEAR 2000 DATE FORMAT WAIVER

Date of Request

System (ACS, ACE, TECS, Administrative, or AES)

Subsystem/Module

Treasury Standard Waiver Requested (Gregorian, Julian, Epoch/Offset or Other)

Customs External Data Exchange Partner(s) (Include impact of variance on their system(s)

Purpose of Data Exchange

Proposed Methodology

Justification for Waiver(Include cost/benefit impact analysis)

Long Term Solution and Implementation Schedule

Signed ___________________ __________ Signed____________________ _________
Project Leader Date System Team Leader Date

For Y2K Project use only

Forwarded to Treasury (name)______________________ date__________________

Follow up date_________________

Appendix J. Vendor Compliance Record

Appendix K. Terms and Definitions

TERMS	DEFINITIONS
Acceptance Testing	Formal testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system. *(CMM and IEEE-610)*
Acceptance Criteria	The criteria that a system or component must satisfy in order to be accepted by a user, customer, or other authorized entity. *(CMM and IEEE-STD-610)*
Action Item	(1) A unit in a list that has been assigned to an individual or group for disposition. (2) An action proposal that has been accepted. *(CMM)*
Action Proposal	A documented suggestion for change to a process or process-related item that will prevent the future occurrence of defects identified as a result of defect prevention activities. *(CMM)*
Action Plan	A set of concrete action tied to Technology Principles and Guidelines for improving Customs IT development, deployment and support. Actions are delineated in the deliverable documents which form the Enterprise IT Architecture Project. *(Cambridge Architecture Process)*
Activity	Any step taken or function performed, both mental and physical, toward achieving some objective. Activities include all the work the managers and technical staff do to perform the tasks of the project and organization. *(CMM)*
Allocated Requirements	The subset of the system requirements that are to be implemented in the software components of the system. The allocated requirements are a primary input to the software development plan and results in software requirements which are documented. *(CMM)*
Application	A computer program designed to help people perform a certain type of work. Depending on the work for which it was designed, an application can manipulate text, numbers, graphics, or a combination of these elements. *(General Industry Term)*

TERMS	DEFINITIONS
Application Domain	A bounded set of related systems (i.e., systems that address a particular type of problem). Development and maintenance in an application domain usually requires special skills and/or resources. Examples include payroll and personnel systems, command and control systems, compilers, and expert systems. *(CMM)*
Application Programming Interface	Application Programming Interface is a routine that expands or contracts data formats during on-line application execution to reconcile expanded or unexpanded data formats between an application and shared data store. *(Year 2000 Software Crisis)*
Application	A major software functional entity. Generally, application and system are synonymous. *(IRS)*
Application Program	A single executable program which consists of source components, such as source modules, JCL, procedures and data files. In some systems, an application program is identified by a run number. (*IRS)*
Architecture Processes	Sets of processes which the IT organization must provide in order to develop, deploy, and support IT applications, products and infrastructure. Architecture Processes are defined and organized across individual activities to provide a more process-oriented approach to IT. *(Cambridge Architecture Process)*
Architecture Framework	The conceptual model for developing an IT architecture. It includes principles, a reference model, and standards. *(Cambridge Architecture Process)*
Architecture	The structure or ordering of components in a computation system. A logical architecture of an application is defined by the class of components and the interrelation of the classes. A physical architecture is determined by the arrangement of the software components and the hardware components. The architecture should be structure to support the Customs business. *(Cambridge Architecture Process)*
Assessment	An appraisal by a trained team of software professionals to determine the state of an organization=s current software process, to determine the high-priority software process-related issues facing an organization, and to obtain the organizational support for software process improvement. *(CMM)*

TERMS	DEFINITIONS
Audit	An independent examination of a work product or set of work products to assess compliance with specification, standards, contractual agreements, or other criteria. *(CMM, IEEE-STD-610)*
Baseline	A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures. *(CMM, IEEE-STD-610)*
Baseline Configuration Management	The establishment of baselines that are formally reviewed and agreed on and serve as the basis for further development. Some software work products, e.g., the software design and the code, should have baselines established at predetermined points, and a rigorous change control process should be applied these items. These baselines provide control and stability when interacting with the customer. *(CMM, IEEE-STD-610)*
Baseline Management	In configuration management, the application of technical and administrative direction to designate the documents and changes to those documents that formally identify and establish baselines at specific times during the life cycle of a configuration item. *(CMM, IEEE-STD-610)*
Benchmark	A standard against which measurements or comparisons can be made. *(CMM, IEEE-STD-610)*
Black Box Testing	A testing method in which the software being tested is treated as an unknown entity. It is based on response to stimuli, so that the tester is looking for a particular result give the type of input that was used. The tester doesn=t have knowledge of what the software is actually doing to the data, only what goes in and what comes out. *(Year 2000 Problem Solver)*
Blueprint	A detailed approach, containing generic and tool-related, step-by-step guidelines, for developing, upgrading, improving, or replacing application systems. Also call a Amethodology@. *(Year 2000 Software Crisis)*
Bridge	A routine that expands or contracts data files in order to reconcile date format differences between expanded and non-expanded date files. Bridges can be applied either during program execution or during file or database conversion. *(IRS)*

TERMS	DEFINITIONS
Business Plan	An action plan that the enterprise will follow on a short-term and/or long-term basis. It specifies the strategic and tactical objectives of U.S. Customs Service over a period of time. The plan, therefore, is time dependent; it will change with the requirements. Although a business plan is usually written in a style unique to U.S. Customs Service, it should concisely describe Awhat@ is planned, Awhy@ it is planned, Awhen@ it will be implemented, by Awho,@ and Ahow@ it will be gauged. The architects of the plan are typically the principals of the Y2K Program. *(General Industry Term)*
Business Function	A group of logically related tasks that are performed together to accomplish an objective. *(General Industry Term)*
Business Area	A functionally distinct corporate division or strategic business unit that is supported by one or more application systems. *(Year 2000 Software Crisis)*
Business Process	The essential business components of U.S. Customs. Each is a collection of activities that take inputs and creates outputs of value to Acustomers@. Customs core business processes are Trade Compliance, Outbound and Passenger. *(Cambridge Architecture Process)*
Business Architecture	A description of the systems, databases, and interactions between systems and databases that will be needed to fulfill business requirements. *(General Industry Term)*
Business Sponsor	Representative from the business organization ultimately responsible for a project. This person is the National Process Owner or Organization Head. *(Customs IT Investment Process)*
Calendar Errors	Errors typically include failing to treat 2000 as a leap year and convening incorrectly between date representations. *(General Industry Term)*
Calendar Date	A representation composed of the time elements: Year, Month-of-Year, and Day-of-Month. The Gregorian calendar is the commonly adopted calendar of business and commerce. There are several format variations for Calendar Date: NIST standard is YYYY-MM-DD; US convention is MM-DD-YYY; European and Military convention is DD-MM-YYYY; and Mon-DD-YY and DD-Mon-YYY are also recognized but less familiar conventions. *(HUD)*

TERMS	DEFINITIONS
Capability Maturity Model (CMM)	A description of the stages through which software organizations evolve as they define, implement, measure, control, and improve their software processes. This model provides a guide for selecting process improvement strategies by facilitating the determination of current process capabilities and the identification of the issues most critical to software quality and process improvement. The CMM was developed at the Software Engineering Institute (SEI) by a group headed up by Watts S. Humphrey (1989) to help organizations improve their software development processes. It ranges from a Level 1 (ad hoc) to a Level 5 (optimizing). *(CMM)*
Causal Analysis Meeting	A meeting, conducted after completing a specif task, to analyze defects uncovered during the performance of that task. *(CMM)*
Causal Analysis	The analysis of defects to determine their underlying root cause. *(CMM)*
CCYY Format	A 4-digit-year format that uses two century digits (CC) to indicate the century and two year digits (YY) to indicate the year within the century. *(HUD)*
CCYYDDD Format	A seven-digit-year format that uses two century digits (CC) to indicate the century, two year digits (YY) to indicate the year within the century, and three digits (DDD) to indicate the day. *(HUD)*
Century Midpoint	An application parameter representing the 2-digit year upon which the century date will toggle in a Year 2000 procedural work around project. Any 2-digit year higher than the century midpoint would be assumed to have a A19" in the century field, and those lower than or equal to the century midpoint would be assumed to have a A20" in the century field. *(Year 2000 Software Crisis)*
Century Date Change	The impending Year 2000 millennium change and its resultant impact and potential disruptions of business services, information databases, and application systems. Same as Year 2000. *(General Industry Term)*

TERMS	DEFINITIONS
Century Rollover Test	A test that can be run on most personal computers, to get an idea of what will happen when the date actually changes to the Year 2000. Before you start your test, make sure that your data files are backup to disk, and make sure that you don=t have your computer connected to a network. Set your computer clock to December 31, 1999, at 12:58 p.m. Turn your computer off, and with three minutes. When you turn the computer back on, check the time and date. Unless you have a fairly new computer-and not all of them are compliant-you will probably see some date other than January 1, 2000. *(Year 2000 Problem Solver)*
Century-Date Compliant	Status of a given unit of software if it correctly handles comparisons, calculations, and sorts that use date data spanning both sides of the century boundary. The goal of a Year 2000 initiative is to achieve century-date compliance for all the software in the enterprise portfolio. *(HUD)*
Certification	The process of verifying a given application or set of programs to determine if they correctly handle century dates. This process must be performed before assuming any application is century-date compliant. Certification is performed before migration activities for applications that are thought to be century-date compliant and is the final step of the validation process for noncompliant applications that have been migrated. *(Year 2000 Software Crisis)*
Certification (of systems)	Describes the state of your organization=s systems once they have achieved compliance and been verified by key personnel via a signature of the proper documents. *(Department of Treasury)*
Certified System	Certified system is a system which the system administrator has signed off on as compliant via the Year 2000 Checklist Evaluation Criteria. *(General Industry Term)*
Commitment	A pact that is freely assumed, visible, and expected to be kept by all parties. *(CMM)*

TERMS	DEFINITIONS
Compliance	Describes the state of your organization=s systems when they can accurately function and process date data from, into and between centuries (in specific the twentieth and twenty-first century) including the years 1999 and 2000 and can accurately exchange data with eternal entities according to the Treasury standard that was mandated in the memorandum from Mr. Flyzik dated June 3, 1997. Processing of date data includes but is not limited to calculations, sequencing, and comparison and pertains to hardware, software and non-it systems. *(General Industry Term)*
Compliant	ACompliant@ system=s hardware and software products correctly process date and date related data individually and in combination in both the 20th and 21st Centuries. ACompliant@ system=s dates are stored, manipulated (including, but not limited to calculating, comparing and sequencing), exchanged, and displayed in a way that cannot be misinterpreted and are not ambiguous. Finally, Acompliant@ systems have no extended semantics, calendar errors, date overflow, and inconsistent semantics. *(General Industry Term)*
Component	A single resource with defined characteristics. The component concept is used in defining precise specifications for testing the validity of various resources. These components are also defined by their relationship to other components. *(HUD)*
Configuration Item (CI)	An aggregation of hardware, software, or both, that is designated for configuration management and treated as a single entity in the configuration management process. *(CMM, IEEE-STD-610)*
Configuration Control	An element of configuration management, consisting of the evaluation, coordination, approval or disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification. *(CMM, IEEE-STD-610)*
Configuration Management (CM)	A discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item, control changes to those characteristics, record and report change processing and implementation status, and verify compliance with specified requirements (i.e., the continuous control of changes made to a system=s hardware, software, and documentation throughout the development and operational life of the system). *(General Industry Term)*

TERMS	DEFINITIONS
Configuration Management Library System	The tools and procedures to access the contents of the software baseline library. *(CMM)*
Configuration Unit	The lowest level entity of a configuration item or component that can be placed into, and retrieved from, a configuration management library system. *(CMM)*
Configuration Identification	An element of configuration management consisting of the evaluation, coordination, approval or disapproval, and establishment of their configuration identification. *(CMM, IEEE-STD-610)*
Conformity	Year 2000 conformity means that neither performance nor functionality is affected by dates, prior to, during and after the year 2000. No value for current date will cause any interruption in operation. Date-based functionality must behave consistently for dates prior to, during and after year 2000. In all interfaces and data storage, the century in any date must be specified either explicitly or by unambiguous algorithms or inferencing rules. Year 2000 must be recognized as a leap year. *(General Industry Term)*
Consistency	The degree of uniformity, standardization, and freedom from contradiction among the documents or parts of system or component. *(CMM, IEEE-STD-610)*
Contingency Factor	An adjustment (increase) of a size, cost, or schedule plan to account for likely underestimates of these parameters due to incomplete specification, inexperience in estimating the application domain, etc. *(CMM)*
Contingency Plan	A plan for responding to the loss of system use due to a disaster such as a flood, fire, computer virus, or major software failure. The plan contains procedures for emergency response, backup, and post-disaster recovery. *(General Industry Term)*
Contract Terms and Conditions	The stated legal, financial, and administrative aspects of a contract. *(CMM)*
Conversion	The process of making changes to databases or source code. *(HUD)*

TERMS	DEFINITIONS
COTS	Commercial Off-the-Shelf. Can refer either to software or hardware. Any package that is purchased commercially, and not contracted for or developed in-house. Microsoft Word is an example of a COTS package. Most compilers are COTS. *(Year 2000 Problem Solver)*
Critical Computer Resource	The parameters of the computing resources deemed to be a source of risk to the project because the potential need for those resources may exceed the amount that is available. Examples include target computer memory and host computer disk space. *(CMM)*
Critical Path	A series of dependent tasks for a project that must be completed as planned to keep the entire project on schedule. (*CMM)*
Customer	The individual or organization that is responsible for accepting the product from the developing organization. *(CMM)*
Data Dictionary	A set of data descriptions that can be shared by several applications. *(HUD)*
Data Compression	Algorithms are used to compress numbers into a tighter space than is needed to hold human readable values. The numbers are expressed in a form that is understandable by the software and computer, but not easily understandable by the average human. This would allow the eight digits in the date field to be fit into the same space that was originally occupied by six digits, as long as the date fields are not already using data compression. This is a form of a logic fix. *(Year 2000 Problem Solver)*
Data Name Rationalization	The process by which data element definitions are modified so that each element retains the same name and characteristics throughout an application system. This process has a tendency to dramatically reduce the actual number of record groups and physical data names within a system by creating reusable Copy or Include code blocks for a given record, segment, or table definition. *(Year 2000 Software Crisis)*
Data Overflow	Many software products represent dates internally as a base date/time plus an offset in days, seconds, or microseconds since that base date/time. Hardware integers holding the offset value can overflow past the maximum corresponding date-an event that may lead to undefined behaviors. *(General Industry Term)*

TERMS	DEFINITIONS
Database	An aggregation of data; a file consisting of a number of records or tables, each of which is constructed of files of a particular type, together with a collection of operations that facilitate searching, sorting, recombination, and similar operations. *(General Industry Term)*
Date Integrity	This criterion primarily covers the correctness of manipulations of date data. These manipulations need to be reliable only over the range of dates that an application is expected to handle. For example, sales-order processing may handle dates from 5 years in the past to one year in the future. In contrast, an employee database may store dates of birth from early in the 20th century to planned retirement dates will into the 21st century. *(General Industry Term)*
Day-of-Month	Is represented by the ordinal numbers 01 through 31, representing the first through the thirty-first days. *(HUD)*
Day-of-Week	Can be expressed as the leading one or leading three characters in the names of the days of the week. *(HUD)*
Day-of-Year	Is represented by the ordinal numbers ranging from 001 (January 1) through 365 (or 366 in Leap Years) for December 31. *(HUD)*
Debug	With software, to detect, locate, and correct logical or syntactical errors in a computer program. *(HUD)*
Decision Support Team (DST)	Subordinate working group of the IRB responsible for validating and scoring project information against standard investment rating criteria and submitting the results of their analyses to the IRB for selection, control, and evaluation decisions. *(Customs IT Investment Process)*
Defect Prevention	The activities involved in identifying defects or potential defects and preventing them from being introduced into a product. *(CMM)*
Defect	A flaw in a system or system component that causes the system or component to fail to perform its required function. A defect, if encountered during execution, may cause a failure of the system. *(CMM)*
Defect Root Cause	The underlying reason (e.g., process deficiency) that allowed a defect to be introduced. *(CMM)*

TERMS	DEFINITIONS
Defect Density	The number of defects identified in a product divided by the size of the product component (expressed in standard measurement terms for that product). *(CMM)*
Dependency Item	A product, action, piece of information, etc., that must be provided by one individual or group to a second individual or group so that the second individual or group can perform a planned task. *(CMM)*
Developmental Configuration Management	The application of technical and administrative direction to designate and control software and associated technical documentation that define the evolving configuration of a software work product during development. Developmental configuration management is under the direct control of the developer. Items under developmental configuration management are not baselines, although they may be baselined and placed under baseline configuration management at some point in their development. *(CMM)*
Deviation	A noticeable or marked departure from the appropriate norm, plan, standard, procedure, or variable being reviewed. *(CMM)*
Effective Process	A process that can be characterized as practiced, documented, enforced, trained, measured, and able to improve. *(CMM)*
End User	The individual or group who will use the system for its intended operational use when it is deployed in its environment. *(CMM)*
Engineering Group	A collection of individuals (both managers and technical staff) representing an engineering discipline. Examples of engineering disciplined include systems engineering, hardware engineering, system test, software engineering, software configuration management, and software quality assurance. *(CMM)*
Environment	The applications, infrastructure, standards, and process that together support an organization=s computation needs. A production environment consists of all the application, procedures and processes for supporting an organization=s business and the associated infrastructure. A development environment consists of all the end-user tools, standards, procedures, processes, methods, and infrastructure to support the development of application. *(Cambridge Architecture Process)*
Event Horizon	The amount of time before a Year 2000 migration effort for a particular application must begin. *(Year 2000 Software Crisis)*

TERMS	DEFINITIONS
Event Horizon	The date by which renovation has to begin to avoid application failure. It is determined by ensuring there is enough renovation time remaining before the failure date. *(HUD)*
Executable lines of code	Source lines of code minus comments, white-space and data declarations. The unit of measure used in costing models to capture the effort to create the function portion of a software program. Used to cost out the effort to develop the functionality. *(General Industry Term)*
Explicit Century	This criterion essentially requires the capability to store explicit values for century. For example, third-party products that use a 4-digit year in all date data elements stored and passed across each interface (including the user interface) would satisfy this criterion. A base-and-offset representation of dates that covers all centuries of interest would also satisfy this criterion. *(General Industry Term)*
Extended Semantics	In general, specific values for a date field are reserved for special interpretation. The most common example is interpreting A99" in a 2-digit year field as an indefinite end date, i.e., Adoes not expire@. Another is embedding a date value in a non-date data element. *(General Industry Term)*
Extensibility	A property of software such that new kinds of components or functionality can be added to it with little or no effort to the existing system. *(Cambridge Architecture Process)*
External System File	A file or date store received from or passed to an interface system. *(Year 2000 Software Crisis)*
Failure Time Horizon	The Failure Time Horizon is the amount of time prior to January 1, 2000, when a component will begin to produce errors in data processing logic. *(IRS)*
Failure Date	The first date on which the application will fail. This depends on the range of dates which have to be handled and any special coding techniques. *(HUD)*
Findings	The conclusions of an assessment, evaluation, audit, or review that identify the most important issues, problems, or opportunities within the area of investigation. *(CMM)*

TERMS	DEFINITIONS
Five-step Approach	The five-step approach is a simple, high-level methodology for attacking the Year 2000 Problem. The five steps are: awareness step, assessment step, renovation step, validation step, and implementation step. *(Year 2000 Problem Solver)*
Fixed Window	A technique to determine the century (high order digits) of a year when represented by two digits. The 2-digit year is compared against a hard coded threshold. The century designation is limited to a 100 year-year range spanning only two centuries. For example, assume the threshold is 60, then if the 2-digit year is greater than or equal to 60, the year is in the 20th century; if the 2-digit is less than 60, the year is in the 21st century. *(HUD)*
Function	A set of related actions, undertaken by individuals or tools that are specifically assigned or fitted for their roles, to accomplish a set purpose or end. *(CMM)*
General Integrity	As a system date advances normally on a host processor, each date roll-over must not lead either the host process nor any software executing there to erroneous processing. The best recognized, high-risk, date change is the roll-over to 2000. *(General Industry Term)*
Goal, Question, Metrics Paradigm	The Goal, Question, Metrics (GQM) paradigm is a method of defining useful and meaningful metrics (measurements) in software, and was developed by Dr. Victor Basili (1989). *(Year 2000 Problem Solver)*
Gregorian Calendar	This is the calendar that is currently being used by most of the world. It was put into place by Pope Gregory in 1582. *(Year 2000 Problem Solver)*
Group	The collection of departments, managers, and individuals who have responsibility for a set of tasks or activities. A group could vary from a single individual assigned part time, to several part-time individuals assigned from different departments, to several individuals dedicated full time. *(CMM)*
I/O Record Group	A record group that contains at least one record definition used in an input/output transaction. *(Year 2000 Software Crisis)*

TERMS	DEFINITIONS
Implicit Century	This last criterion requires that, if the century is not explicitly provided, its value can be correctly inferred with 100% accuracy from the value of date provided. For example, the range of values for an Ainvoice date@ would rarely span more than 10 years. Because the century can always be guessed correctly for an invoice date with a 2-digit year, this date data element would satisfy this criterion. Note that this criterion permits cost-risk trade-offs that minimize changes to existing date formats. *(General Industry Term)*
Inconsistent Semantics	At an interface between systems, software on each side assumes semantics of data passed. Software must make same century assumptions about 2-digit years. *(General Industry Term)*
Incremental Development	A strategy for developing systems based on making progress in small steps. For software, a system is constructed in parts rather than attempting to build the entire system at one time. *Cambridge Architecture Process)*
Information Technology (IT) Portfolio	A collection of IT projects, including applications and other technical investments, that represents the mixture of projects that best meet the mission needs of the agency. *(Customs IT Investment Process)*
Information Technology Projects	An IT Project is any item that is focused on developing and/or maintaining a specific IT product, which may include hardware, software, and other components. Products typically are organized for the collection, process, maintenance, transmission, and dissemination of information. (*Customs IT Investment Process)*
Information Architecture	A description of the enterprise in terms of its business activity, business information, and their interaction. *(General Industry Term)*
Information Technology	Information Technology maximizes the use of Information and applied Technology to enhance Productivity and meet the Challenges of the U.S. Customs Mission. U.S. Customs is to become an information-based agency that maximizes the use of technology to achieve mission effectiveness and resource efficiency. *(General Industry Term)*
Information System	An Information System is a collection of people, procedures, and equipment designed, built, operated, and maintained to collect, record, process, store, retrieve and display information. *(Encyclopedia of Computer Science, 3rd Edition)*

TERMS	DEFINITIONS
Infrastructure	The collection of hardware, networks, software services tools, and services that support the development and execution of applications. *(Cambridge Architecture Process)*
Integrated Software Management	The unification and integration of the software engineering and management activities into a coherent defined software process based on the organization=s standard software process and related process assets. *(CMM)*
Integration	Two or more software applications that must run on the same physical processor(s) and under the same operating system. *(General Industry Term)*
Integration Testing	Testing to determine that the related information system components perform to specification. *(General Industry Term)*
Interface	A boundary across which two systems communicate. An interface might be a hardware connector used to link to other devices, or it might be a convention used to allow communication between two software systems. This is to include interfaces internal to the system, its applications and programs, to other internal or external systems. *(General Industry Term)*
Interface System	Any system or conversion unit that receives data from or sends data to another system or conversion unit currently undergoing an assessment or conversion effort. *(IRS)*
Interoperability	The ability of two or more systems or components to exchange data and use information. The ability of two or more systems to exchange information and to mutually use the information that has been exchanged. *(General Industry Term)*
Inventory	In the context of a year 2000 program, the process of determining the components that comprise the agency=s systems portfolio. The inventory should include all applications, databases, files, and related system components that will require inspection to locate date data and related date computations. *(HUD)*
Investment Review Board (IRB)	The executive management team at the U.S. Customs Service that makes funding decisions based upon comparisons and tradeoffs among competing project proposals. The IRB plays a critical role in the funding process, especially when evaluating those projects expected to have an organizational-wide impact. *(Customs IT Investment Process)*

TERMS	DEFINITIONS
Investment Management Process (IMP)	The framework within which information technology decisions at Customs are made. This definable and repeatable process must be linked to other management and planning cycles. *(Customs IT Investment Process)*
Iterative Development	A strategy for developing systems that allows for the controlled reworking of part of a system to remove mistakes or to make improvements based on user feedback. *(Cambridge Architecture Process)*
Julian Date	A date in format CCYYDDD. A date format that contains the year in positions 1 and 4, and the day in positions 5 through 7. The day is represented as 1 through 366, right adjusted, and padded with zeroes on the left. *(HUD)*
Junction	Interface point where a system with expanded fields passes data, receives data, or otherwise interacts with another system whose corresponding fields have remained unchanged. *(Year 2000 Software Crisis)*
Key Process Area	A cluster of related activities that, when performed collectively, achieve a set of goals considered important for establishing process capability. The key process areas have been defined to reside at a single maturity level. They are the areas identified by the Software Engineering Institute (SEI) to be the principal building blocks to help determine the software process capability of an organization and understand the improvements needed to advance to higher maturity levels. *(CMM)*
Key Personnel	This should be a list of people within each organization who have responsibility for a particular system including but not limited to: programmer, program supervisor, system owner, system user, and the executive in charge of the Year 2000 Project. *(Department of Treasury)*
Key Business Drivers	High-level, general factors which place demands on a business organization or business unit to perform. *(Cambridge Architecture Process)*

TERMS	DEFINITIONS
Leap Year Problem	The leap year problem comes into play when the programmer who developed the date routines only knew two of the three rules governing how to determine whether a year is a leap year or not. The rules are a year is a leap year if it is divisible by four, unless it is divisible by 100; but if it is divisible by 400, then it is a leap year. So between 1600 and 2100, every fourth year is a leap year except 1700, 1800, 1900, and 2100. *(Year 2000 Problem Solver)*
Leap Year	Leap Year is a year with an additional day (YYYY-02-29). Leap years occur in all years evenly divisible by 400 or in years evenly divisible by 4 and not evenly divisible by 100. For example the year 1900 was not a leap year but the year 2000 is a leap year. *(HUD)*
Legacy Transition Meta-Model	An entity relationship model representing current and target system components and relationships among those components. It is usually customized for a given project and requires an open repository to support practical use. It is used to cross-reference, trace, track, and audit physical, logical, and externally defined system components before, during, and after a redevelopment project. *(Year 2000 Software Crisis)*
Legacy Systems	Systems that are already in existence and under maintenance, as opposed to being under current development. (*Year 2000 Problem Solver)*
Legacy Code	Code that is being used in the legacy systems. A good portion of this code is being enhanced or modified on a fairly regular basis, but some of it is so old, and written in a language that is so obsolete, that no one knows how to modify it. For years some of this code has been running along not causing any problems for anyone. But now that we are facing the Year 2000, a lot of it will malfunction. *(Year 2000 Problem Solver)*
Lilian Date	The number of days since 14 October 1582. 15 October 1582 is Lilian Day 1, 16 October 1582 is Lilian day 2, and so on. (Names for Aloysius Lilus - an advisor to Pope Gregory XIII - who, together with his brother, constructed the current Gregorian calendar). *(HUD)*
Line of Code (LOC)	A single computer program command, declaration, or instruction. Program size is often measured in line of code. *(General Industry Term)*

TERMS	DEFINITIONS
Local Data Element	A data element that accomplishes an intermediate task or calculation within a program and only within that program. (*Year 2000 Software Crisis)*
Logical Record	The full aggregation of attributes with no redundancy describing an I/O record in the context of the programming language being used. There can be many physical records used to describe a single logical record. *(Year 2000 Software Crisis)*
Logical Fix	Correcting the Year 2000 Problem with some sort of algorithm so the two-digit year fields don=t have to be replaced within the software and databases. An example of a logical fix is using the value 50 as the cutoff number for dates, so if a number is less than 50, the date is 2000 plus that number. If the number is 50 or greater, then the date is 1900 plus that number. *(Year 2000 Problem Solver)*
Maintenance	The process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment. *(CMM, IEEE-STD-610)*
Managed and Controlled	The process of identifying and defining software work products that are not part of a baseline and, therefore, are not placed under configuration management but that must be controlled for the project to proceed in a disciplined manner. AManaged and controlled@ implies that the version of the work product in use at a given time (past or present) is known (i.e., version control), and changes are incorporated in a controlled manner (i.e., change control). *(CMM)*
Measure	A unit of measurement (such as source lines of code or document pages of design). *(CMM)*
Measurement and Analysis	A description of the need to measure the process and analyze the measurements. Measurement and Analysis typically includes examples of the measurements that could be taken to determine the status and effectiveness of the Activities Performed. *(CMM)*
Measurement	The dimension, capacity, quantity, or amount of something (e.g., 300 sources lines of code of 7 document pages of design). *(CMM)*
Method	A reasonable complete set of rules and criteria that establish a precise and repeatable way of performing a task and arriving at a designed result. *(CMM)*

TERMS	DEFINITIONS
Methodology	A collection of methods, procedures, and standards that defines an integrated synthesis of engineering approaches to the development of a product. *(CMM)*
Methodology	A detailed approach, containing generic and tool-related, step-by-step guidelines, to developing, upgrading, improving, or replacing application systems. Also called a Ablueprint@. *(Year 2000 Software Crisis)*
Metrics	Means by which software engineers measure and predict aspects of processes, resources, and products that are relevant to the software engineering activity. *(HUD)*
Migration System	An existing automated information system (AIS) or application, or a planned and approved AIS or application, that has been officially designated as the single AIS or application to support standard processes for a function. *(General Industry Term)*
Milestone	A scheduled event for which some individual is accountable and that is used to measure progress. *(CMM)*
Mission Critical System	(1) A system supporting a core business activity or process. (*GAO Year 2000 Computing Crisis: An Assessment Guide)* (2) A system that when its capabilities are degraded, the organization realizes a resulting loss of a core capability. *(DOD Year 2000 Management Plan V.1.0, dated April 1997)*
Mission	The essential top level reason for being of U.S. Customs. *(Cambridge Customs Technology Architecture Process Paper.)*
Nontechnical Requirements	Agreements, conditions and/or contractual terms that affect and determine the management activities of a software project. *(CMM)*
Object code	The machine code generated by a source code language processor such as an assembler or compiler. A file of object code may be immediately executable or it may required linking with other object code files, e.g., libraries, to produce a complete executable program. *(General Industry Term)*
Open System	A comprehensive set of interfaces, services and supporting formats to achieve Interoperability, portability, and interaction. *(Cambridge Architecture Process)*

TERMS	DEFINITIONS
Operating System	The software which schedules tasks, allocates storage, handles the interface to peripheral hardware, and presents a default interface to the user when no application program is running. *(General Industry Term)*
Operational Alignment	The link between business infrastructure and processes and IT infrastructure and processes. It is the match between business requirements and expectation and IT=s delivery capability. *(Cambridge Architecture Process)*
Operational Software	The software that is intended to be used and operated in a system when it is delivered to its customer and deployed in its intended environment. *(CMM)*
Ordinal Date	A representation composed of the time elements Year and Day-of-Year. This representation is also referred to as the Julian Date. *(HUD)*
Organization	A unit within a company of other entity (e.g., government agency or branch of service) within which many projects are managed as a whole. All projects within an organization share a common top-level manager and common policies. *(CMM)*
Organization=s Measurement Program	The set of related elements for addressing an organization=s measurement needs. It includes the definition of organization-wide measurements, methods and practices for collecting organization measurement data, methods and practices for analyzing organizational measurement data, and measurement goals for the organization. *(CMM)*
Organization=s Software Process Assets	A collection of entities, maintained by an organization, for use by projects in developing, tailoring, maintaining, and implementing their software processes. These software process assets typically include: the organization=s standard software process; descriptions of the software life cycle approved for use; the guidelines and criteria for tailoring the organization=s standard software process; the organization=s software process database, and a library of software process-related documentation. Any entity that the organization considers useful in performing the activities of process definition and maintenance could be included as a process asset. *(CMM)*

TERMS	DEFINITIONS
Organization=s Software Process Database	A database established to collect and make available data on the software processes and resulting software work products, particularly as they relate to the organization=s standard software process. The database contains or references both the actual measurement data and the related information needed to understand the measurement data and assess it for reasonableness and applicability. Examples of process and work product data include estimates of software size, effort, and cost; actual data on software size effort, and cost; productivity data; peer review coverage and efficiency; and number and severity of defects found in the software code. *(CMM)*
Outsourcing	Paying another company to provide services which an organization might otherwise have performed itself, e.g. software development. *(General Industry Term)*
Parallel Processing	The simultaneous use of more than one computer to solve a problem. *(General Industry Term)*
Pareto Analysis	The analysis of defects by ranking causes from most significant to least significant. Pareto analysis is based on the principle, named after the 19th-century economist Wilfredo Pareto, that most effects come from relatively few causes, i.e., 80% of the effects come from 20% of the possible causes. *(CMM)*
Parsing	The conversion of physical system components into data structures suited to code transformations within a software engineering tool. All tools used to document, analyze, import, reverse, or in any other way process existing source of executable object types of any nature require this facility. An example of a tool containing parsing technology is a source code compiler. *(Year 2000 Software Crisis)*
Peer Review	A review of a software work product, following defined procedures, by peers of the produces of the product for the purpose of identify defects and improvements. *(CMM)*
Physical Fix	Correcting the Year 2000 Problem by actually changing all of the two-digit year fields in the software and databases to four-digit year fields. (*Year 2000 Problem Solver)*
Physical System Components	The actual pieces that constitute an executable production system. Examples are JCL, load modules, source code, screen maps, and related items. *(Year 2000 Software Crisis)*

TERMS	DEFINITIONS
Platform	The foundation technology of a computer system. Typically, a specific combination of hardware and operating system. *(General Industry Term)*
Policy	A guiding principle, typically established by senior management, which is adopted by an organization or project to influence and determine decisions. *(CMM)*
Portability	The ability to implement and execute software in one type of computing space and have it execute in a different computing space with little or no changes. *(Cambridge Architecture Process)*
Portfolio	In the context of the year 2000 program, an inventory-preferably automated-of an agency=s information systems and their components grouped by business areas. *(General Industry Term)*
Principles	Principles are the enduring, abstract or high-level rules and objectives which govern IT within U.S. Customs. *(Cambridge Architecture Process)*
Procedure	A written description of a course of action to be taken to perform a given task. *(CMM, IEEE-STD-610)*
Process	The activities and methods used to build products or support a business. *(Cambridge Architecture Process)*
Process Performance Baseline	A documented characterization of the actual results achieved by following a process, which is used as a benchmark for comparing actual process performance against expected process performance. A process performance baseline is typically established at the project level, although the initial process performance baseline will usually be derived from the process capability baseline. *(CMM)*
Process Model	Collections of maxims, strategies, activities, methods, and tasks that are organized to achieve a set of goals and objectives. *(Cambridge Architecture Process)*
Process Description	The operational definition of the major components of a process. Documentation that specifies, in a complete, precise, verifiable manner, the requirements, design, behavior, or other characteristics of a process. It may also include the procedures for determining whether these provisions have been satisfied. Process descriptions may be found at the task, project, or organizational level. *(CMM)*

TERMS	DEFINITIONS
Process Capability Baseline	A documented characterization of the range of expected results that would normally be achieved by following a specific process under typical circumstances. A process capability baseline is typically established at an organization level. *(CMM)*
Production Environment	The system environment where the agency performs its routine information process activities. *(General Industry Term)*
Program	An executable load module that is made up of two or more application modules which when integrated, accomplish a defined task, a program can consist of a single module which calls only general utility modules. This would not be considered modular and would not require program integration testing, but would require system integration testing. *(IRS)*
Proper Documents	This refers to a form that will need to be drafted for each bureau and will include all software, hardware and non-IT systems. This form will be the baseline used in order to track certification. Each system will also have a compliance form which will contain a checklist of items that will verify the compliance of the system. These items will contain the following information: what code was changed, when the code was changed, by whom was the code changed, the number of LOC changed, sign-off that unit test has been completed successfully, sign-off of successful completion of performance test, and a statement of system impact due to conversion (if there is an impact on a particular system, an explanation will need to be provided). *(Department of the Treasury)*
Prototype	A preliminary, or intentionally incomplete or scaled-down version of a system. *(Cambridge Architecture Process)*
Quality Assurance	In the case of Year 2000, the critical review process required to ensure that a century-date compliance task was adequately and correctly performed, existing functionality was not unintentionally altered, and century-date compliance was achieved. This process includes monitoring coding quality, performing validation tests, and measuring compliance to standards. All the planned and systematic actions necessary to provide adequate confidence that a product or service will satisfy given requirements for quality. *(Year 2000 Software Crisis)*

TERMS	DEFINITIONS
Quality	(1) The degree to which a system, component, or process meets specified requirements. (2) The degree to which a system, component, or process meets customer or user needs or expectations. *(CMM, IEEE-STD-610)*
Quantitative Control	Any quantitative or statistically-based technique appropriate to analyze a software process, identify special causes of variations in the performance of the software process, and bring the performance of the software process within well-defined limits. *(CMM)*
Record Group	A group of elements (such as a COBOL 01 level) or I/O records that have been grouped together according to common lengths, underlying record structure, or verb transfer usage (e.g., Move, Read Into). *(Year 2000 Software Crisis)*
Regression Testing	Selective retesting to detect faults introduced during modification of a system. *(GAO)*
Renovation	The process of making the changes needed to ensure the successful operation of the application for future dates. *(HUD)*
Repository	An information storage facility or central database that contains all meta-data relevant to the management, design, implementation, and transition of one or more information systems or for the enterprise. *(Year 2000 Software Crisis)*
Risk Assessment	A continuous process performed during all phases of system development to provide an estimate of the damage, loss, or harm that could result from a failure to successfully develop individual system components. *(General Industry Term)*
Risk	Possibility of suffering loss. Risk management includes risk identification, analysis, prioritization, and control. *(CMM)*
Risk Management	(1) A management approach designed to reduce risks inherent to system development. *(GAO)* (2) Risk management includes risk identification, analysis, prioritization, and control. *(CMM)*
Scalability	The likelihood that an artifact can be extended to provide additional functionally with little or no additional effort. *(Cambridge Architecture Process)*

TERMS	DEFINITIONS
Scenario	A project-oriented, assessment, and implementation work plan template that incorporates required blueprint/methodology steps into pre-packages solutions to address century-date migrations. Scenarios provide a means for organizations with little or no experience in large-scale migrations to rapidly deploy methodology techniques for their century-date compliance projects. *(Year 2000 Software Crisis)*
Sliding Window	A form of a logical fix in which an arbitrary number is selected as the cutoff number for dates, such as 50, but each year the software increases that number. So in 2000 the cutoff is 50, in 2001 the cutoff is 51, so on. *(Year 2000 Problem Solver)*
Software Process Maturity	The extent to which a specific process is explicitly defined, managed, measured, controlled, and effective. Maturity implies a potential for growth in capability and indicates both the richness of an organization=s software process and the consistency with which it is applied in projects throughout the organization. *(CMM)*
Software Product	The complete set, or any of the individual items of the set, of computer programs, procedures, and associated documentation and data designated for delivery to a customer or end user. *(CMM, IEEE-STD-610)*
Software Process Element	A constituent element of a software process description. Each process element covers a well-defined, bounded, closely related set of tasks (e.g., software estimating element, software design element, coding element, and peer review element). The descriptions of the process elements may be templates to be filled in, fragments to be completed, abstractions to be refined, or complete descriptions to be modified or used unmodified. *(CMM)*
Software Project	An undertaking requiring concerted effort, which is focused on analyzing, specifying, designing, developing, testing, and/or maintaining the software components and associated documentation of a hardware/software system. *(CMM)*
Software Framework	A set of components that define the basic abstractions of a particular set of applications or of a domain. A software framework must be refined with application-specific details in order to be used. *(Cambridge Architecture Process)*

TERMS	DEFINITIONS
Software Process Description	The operational definition of a major software process component identified in the project=s defined software process or the organization=s standard software process. It documents, in a complete, precise, verifiable manner, the requirements, design, behavior, or other characteristics of a software process. *(CMM)*
Software Development Environment	Collection of methods, languages, libraries, tools, and database services needed to create software systems. *(Cambridge Architecture Process)*
Software Reengineering	The use of tools and techniques to facilitate the analysis, improvement, redesign, and reuse of existing software systems to support changing business and technical information requirements. *(Year 2000 Software Crisis)*
Software Process Assessment	An appraisal by a trained team of software professionals to determine the state of an organization=s current software process, to determine the high-priority software process-related issues facing an organization, and to obtain the organizational support for software process improvement. *(CMM)*
Software Process	A set of activities, methods, practices, and transformation that people use to develop and maintain software and the associated products (e.g., project plans, design documents, code, test cases, and user manuals). *(CMM)*
Systems Development Life Cycle (SDLC)	U.S. Customs SDLC is the development methodology that should be used by all projects. *(Customs IT Investment Process)*
Software Quality Assurance	(1) A planned and systematic pattern of all actions necessary to provide adequate confidence that a software work product conforms to established technical requirements. (2) A set of activities designed to evaluate the process by which software work products are developed and/or maintained. *(CMM)*
Software Baseline Audit	An examination of the structure, contents, and facilities of the software baseline library to verify that baselines conform to the documentation that describes the baselines. *(CMM)*
Software Baseline Library	The contents of a repository for storing configuration items and the associated records. *(CMM)*
Software Build	An operation version of a software system or component that incorporates a specified subset of the capabilities the final software system or component will provide. *(CMM, IEEE-STD-610)*

TERMS	DEFINITIONS
Software Configuration Control Board	A group responsible for evaluating and approving or disapproving proposed changes to configuration items, and for ensuring implementation of approved changes. *(CMM)*
Software Engineering Group	The collection of individuals (both managers and technical staff) who have responsibility for software development and maintenance activities (i.e., requirements analysis, design, code, and test) for a project. Groups performing software-related work, such as the software quality assurance group, the software configuration management group, and the software engineering process group, are not included in the software engineering group. *(CMM)*
Software Work Product	Any artifact created as part of defining, maintaining, or using a software process, including process descriptions, plans, procedures, compute programs, and associated documentation, which may or may not be intended for delivery to a customer or end user. *(CMM)*
Software Requirement	A condition or capability that must be met by software needed by a user to solve a problem or achieve an objective. *(CMM, IEEE-STD-610)*
Software Engineering Process Group	A group of specialists who facilitate the definition, maintenance, and improvement of the software process used by the organization. In the key practices, this group is generically referred to as Athe group responsible for the organization=s software process activities@. *(CMM)*
Software Quality Management	The process of defining quality goals for a software product, establishing plans to achieve these goals, and monitoring and adjusting the software plans, software work products, activities, and quality goals to satisfy the needs and desires of the customer and end users. *(CMM)*
Software Engineering Staff	The software technical people (e.g., analysts, programmers, and engineers), including software task leaders, who perform the software development and maintenance activities for the project, but who are not managers. *(CMM)*
Software-related Group	A collection of individuals (both managers and technical staff) representing a software engineering discipline that supports, but is not directly responsible for, software development and/or maintenance. Examples of software engineering disciplines include software quality assurance and software configuration management. *(CMM)*

TERMS	DEFINITIONS
Source Code	The form in which a computer program is written by the programmer. Source code is written in a programming language which is then compiled into object code or machine code or executed by an interpreter. *(GAO)*
Stakeholders	Stakeholders include The Year 2000 Project Office, the Business Owners that use the automated systems to execute their business function, the Technical Owners that maintain the automated systems and the Trading Partners that provide input to automated systems or receive output from U.S. Customs automated systems. *(IRS)*
Standard	Mandatory requirements employed and enforced to prescribe a disciplined uniform approach to software development. *(CMM)*
Standard	In computing, a set of detailed technical guidelines used as a means of establishing uniformity in an area of hardware or software development. *(General Industry Term)*
Strategic Goals	The highest level desired result for each Process of Customs. Goals are supported by one or more specific Strategic Objectives. *(Cambridge Architecture Process)*
Strategic Alignment	The link between business strategy and IT strategy. The match between IT=s capability to support and in turn to help shape business strategy. *(Cambridge Architecture Process)*
Strategic Integrated Resources Management (IRM) Plan	A long-term, high-level plan that defines a systematic way of how the agency will use information technology to effectively accomplish the agency=s missions, goals, and objectives. *(General Industry Term)*
Strategic Objectives	The second highest level of desired results for each Process of Customs. One or more objectives support each high-level Goal. *(Cambridge Architecture Process)*
Strategic Intent	The second level reason for being of U.S. Customs. The vision to energize the organization to scope its future direction. *(Cambridge Architecture Process)*
Strategic Plan	A long-term, high-level plan that identifies broad business goals and provides a road map for their achievement. (General Industry Term)

TERMS	DEFINITIONS
Strategy	A collection of ideas that describes and directs the larger vision of how activities in a process model will be carried out, without consideration for the details of the techniques or tasks involved. *(Cambridge Architecture Process)*
Subject Matter Expert (SME)	One who is knowledgeable in the functional or technical aspects of an application system or other area of study, such as redevelopment. *(Year 2000 Software Crisis)*
Synonyms	Data elements that map to the same physical data but have a different definition and a different data name. Synonyms are usually valid redefinitions of data elements and are used for various purposes, such as for reporting and computations. *(Year 2000 Software Crisis)*
System Acceptability Testing	An independent assessment of the quality of the systems software which determines if it satisfies customer specified requirements during a prior development or maintenance modification. *(IRS)*
System Testing	Testing to determine that the results generated by the enterprise=s information systems and their components are accurate and the systems perform to specification. *(GAO)*
System	A collection of components organized to accomplish a specific function or set of functions. *(CMM, IEEE-STD-610)*
System Requirement	A condition or capability that must be met or possessed by a system or system component to satisfy a condition or capability needed by a user to solve a problem. *(CMM, IEEE-STD-610)*
System Engineering Group	The collection of individuals (both managers and technical staff) who have responsibility for specifying the system requirements; allocating the system requirements to the hardware, software, and other components; specifying the interfaces between the hardware, software, and other components; and monitoring the design and development of these components to ensure conformance with their specifications. *(CMM)*
Systems Redevelopment	The process of significantly modifying or rebuilding application system(s) that essentially replaces portions or all of one or more existing systems through the use of software reengineering technology. (*Year 2000 Software Crisis)*
Target Computer	The computer on which delivered software is intended to operate. *(CMM).*

TERMS	DEFINITIONS
Task	(1) A low-level activity that begins and ends. (2) A collection of related steps with a Year 2000 methodology that contains objectives, entrance criteria, rolls/skills, inputs, tools support, deliverables, quality checks, metrics, exit criteria, and generic and tool-based guidelines. *(Year 2000 Software Crisis)* (3) The smallest unit of work on a project schedule. *(Cambridge Architecture Process)* (4) A well-defined unit of work in the software process that provides management with a visible checkpoint into the status of the project. Tasks have readiness criteria (preconditions) and completion criteria (post conditions). *(CMM, IEEE-STD-610)*
Task Leader	The leader of a technical team for a specific task, who has technical responsibility and provides technical direction to the staff working on the task. *(CMM)*
Team	A collection of people, often drawn from diverse but related groups, assigned to perform a well-defined function for an organization or a project. Team members may be part-time participants of the tem and have other primary responsibilities. *(CMM)*
Technical Review Board (TRB)	A group that conducts architecture and standards reviews for al Quarterly Program Reviews. (*Customs IT Investment Process)*
Technical Owner	The Technical Owner is defined as the entity that is responsible for the Technical Operation of that function. *(IRS)*
Technical Requirements	Those requirements that describe what the software must do and its operational constraints. Examples of technical requirements include functional, performance, interface, and quality requirements. *(CMM)*
Technical Architecture Group (TAG)	A group that defines the architecture needed for new IT investments, as well as coordinates the responsibilities of the Technical Review Board (TRB). *(Customs IT Investment Process)*
Technique	The specific details, algorithms, or heuristics employed by a method. *(Cambridge Architecture Process)*
Technology	The application of science an/or engineering in accomplishing some particular result. *(CMM)*

TERMS	DEFINITIONS
Test Facility	A computer system isolated from the production environment dedicated to the testing and validation of applications and system components. *(HUD)*
Test	The process of exercising a product to identify differences between expected and actual behavior. *(General Industry Term)*
Test Facility	A computer system isolated from the production environment dedicated to the testing and validation of applications and system components. *(General Industry Term)*
Testability	(1) The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met. *(CMM, IEEE-STD-610)*
Testing	The process of executing one or more programs in a system to verify that functional capability of that system meets the user requirements specified during a prior development or maintenance modification. *(Year 2000 Software Crisis)*
Tool	A software product that supports one or more century-date compliance project tasks or steps. Tools are general off-the-shelf, commercially available software products that assist with the analysis, upgrade, improvement, or rearchitecting of legacy applications. *(Year 2000 Software Crisis)*
Traceability	The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or master-subordinate relationship to one another. *(CMM, IEEE-STD-610)*
Train	To make proficient with specialized instruction and practice. *(CMM)*
Training Waiver	A written approval exempting an individual from training that has been designated as required for a specific role. The exemption is granted because it has been objectively determined that the individual already possesses the needed skills to perform the role. *(CMM)*

TERMS	DEFINITIONS
Training Group	The collection of individuals (both managers and staff) who are responsible for coordinating and arranging the training activities for an organization. This group typically prepares and conducts most of the training courses and coordinates use of other training vehicles. *(CMM)*
Training Program	The set of related elements that focus on addressing an organization=s training needs. It includes an organization=s training plan, training materials, development of training, conduct of training, training facilities, evaluation of training, and maintenance of training records. *(CMM)*
Unit	(1) A separately testable element specified in the design of a computer software component. (2) A logically separable part of a computer program. (3) A software component that is not subdivided into other components. *(CMM, IEEE-STD-610)*
Unit Testing	Testing to determine that individual program modules perform to specification. (General Industry Term)
Upgrade Unit	A system, or group of related systems, treated as a single unit of work for purposes of providing a redevelopment project scenario. One common application of upgrade until segmentation is for enterprise-wide, century-date change projects. *(Year 2000 Software Crisis)*
Utilities	Computer programs designed to perform maintenance work on the system or on system component-for example, a storage backup program, a disk or file recovery program, or a resource editor. *(HUD)*
Validation	The process for testing the results of a century-date compliance project to ensure their correctness. It is accomplished by processing a series of tests that show (1) the modified applications or programs correctly handle century dates, and (2) existing functionality has not been adversely affected by the project. *(Year 2000 Software Crisis)*
Verification	The process of evaluating software to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. *(CMM, IEEE-STD-610)*

TERMS	DEFINITIONS
Verifying Implementation	The steps to ensure that the activities are performed in compliance with the process that has been established. Verification typically encompasses reviews and audits by management and software quality assurance. *(CMM)*
Well-defined Process	A process that includes readiness criteria, inputs, standards and procedures for performing the work, verification, mechanisms (such as peer reviews), outputs, and completion criteria. *(CMM)*
White Box Testing	Unit testing is a good example of white box testing. The programmer inputs a piece of data into a program, and then follows what path that data takes to go through the program, what decision paths it takes, and what actually happens to the data as it goes through the program. Does it get changed; cause other data to be generated; or just pass through without doing anything? *(Year 2000 Problem Solver)*
Year 2000 Problem	The potential problems and its variations that might be encountered in any level of computer hardware and software from microcode to application programs, files, and databases that need to correctly interpret year-date data represented in 2-digit-year format. *(GAO)*
Year 2000 compliant	Information systems able to accurately process date data - including, but not limited to, calculating, comparing, and sequencing - from, into, and between the twentieth and twenty-first centuries, including leap year calculations. (GAO)
Year 2000 System	An automated process that used information technology such as computer hardware and software to perform a specific function, application, or service. (General Industry Term)
Year 2000 (Y2K)	The impending Year 2000 millennium change and its resultant impact and potential disruptions of business services, information databases, and application systems. Same as Century Date Change. *(Year 2000 Software Crisis)*
Year 2000 Project Office	The core IT project team that coordinates century-date compliance projects across an enterprise. Their responsibilities include communication, skills, transfer, tool management, central infrastructure issues, and third-party relationships. *(Year 2000 Software Crisis)*
Year 2000 Transition	The process of revising all programming entities (programs, databases, and so on) to correctly process date data outside the range 1900-1999. *(HUD)*

ACRONYMS

ABC	Activity Based Costing
ABI	Automated Broker Interface
ACE	Automated Commercial Environment
ACH	Automated Clearinghouse
ACS	Automated Commercial System
ADD	Applications Development Division
ADP	Automated Data Processing
AERP	Automated Export Reporting System
AES	Automated Export System
AIES	Automated Information Exchange System
AIMS	Asset Information Management System
AIS	Automated Information System
AISS	Automated Information System Security
AMS	Automated Manifest System
ANS	American National Standard
ANSI	American National Standards Institute
API	Application Program Interface or Advanced Passenger Information
APIS	Advanced Passenger Information System
APR	Agency Procurement Request
AQIS	Australian Quarantine Information System
ARRS	Automated Receiving Report System
AS	Administration Systems
ATD	Applied Technology Division
ATF	Alcohol, Tobacco, and Firearms
ATS	Automated Targeting System
BAA	Business Area Analysis
BCR	Benefit Cost Ratio
BIA	Business Impact Analysis
BIOS	Basic Input/Output System
BMP	Batch Message Processing
BPI	Business Process Improvement
BPR	Business Processing Re-engineering
BSD	Business System Design
BXA	Bureau of Export Administration
C2	Controlled Access Protection
CAISP	Customs Administrative Services Information Strategy Plan
CALVIN	Customs Advanced Lookout Vehicle Identification Network
CAMIR	Customs Automated Manifest Interface Requirements
CASE	Computer-Aided Software Engineering

CATAIR	Customs And Trade Automated Interface Requirements
CBA	Cost Benefit Analysis
CBT	Computer Based Training
CC	Century indicator
CCB	Configuration Control Board
CDC-2000	Customs Distributed Computing
CDN	Consolidated Data Network
CFO	Chief Financial Officer
CFS	Center for Standards
CI	Configuration Identification
CICS	Custom Information Control System
CIO	Chief Information Officer
CIWG	Communications Interoperability Working Group
CLASS	Consular Lookout and Support System
CLETS	California Law Enforcement Telecommunications System
CM	Configuration Management
CMM	Capability Maturity Model
CMC	Customs Management Center
CMIS	Cost Management Information Systems
CMS	Configuration Management System
COE	Common Operating Environment
COMPEX	Compliance Measurement Statistics
COMSEC	Communications Security
COSS	Customs Overtime Scheduling System
COTHEN	Customs Over-the-Horizon Enforcement Network
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CSF	Critical Success Factors
CSTAP	Computer Security Training and Awareness Program
CTT	Core Technical Team
CUSDEC	Customs Declaration
CUSRES	Customs Response
DASD	Direct Access Storage Device
DBMS	Database management systems
DES	Digital Encryption Standard
DESAC	Disabled Equipment and Services Accommodation Contract
DEA	Drug Enforcement Agency
DOJ	Department of Justice
DOS	Disk Operating System
DOT	Department of Transportation
DPA	Delegation of Procurement Authority
DST	Decision Support Team
EAP	Executive Agent Program
EC	Electronic Commerce
ECASS	Export Control Automated Support System

EC/EDI	Electronic Commerce/Electronic Data Exchange
ECL	Executive Control Language
EDI	Electronic Data Interchange
EDIFACT	Electronic Data Interchange for Administration, Commerce and Transport
EEO	Equal Employment Office
EFT	Electronic Funds Transfer
EIP	Electronic Invoicing Program
EIS	Enterprise Information System
EIT	Executive Improvement Team
ELOC	Executable Line of Code
ELVIS	Electronic Visa Information System
EOAF	Executive Office of Asset Forfeiture
FAA	Federal Aviation Administration
FASAB	Federal Accounting Standards Advisory Board
FBI	Federal Bureau of Investigation
FDA	Food and Drug Administration
FFS	Federal Financial System
FIPS PUB	Federal Information Processing Standards Publication
FITS	Firearms Inventory Tracking System
FMFIA	Federal Managers Financial Improvement Act
FMSAC	Financial Management Systems Advisory Committee
FO	Field Operations
FP&F	Fines, Penalties & Forfeitures
FSN	Field Support Network
FTE	Full-Time Equivalent
FTH	Failure Time Horizon
FTS	Federal Telecommunications System
GAO	General Accounting Office
GES	Global Enrollment System
GMRA	Government Management Reform Act
GOTS	Government off-the-Shelf
GPRA	Government Performance and Results Act (1993)
GQM	Goal, Question, Metrics Paradigm
GSA	General Services Administration
GUI	Graphical User Interface
IAS	Inventory of Automated Systems
IASS	Importer Activity Summary System
IAW	In Accordance With
IBIS	Interagency Border Inspection System
IDEAS	Interior Department Electronic Acquisition System
IE	Information Engineering
IEEE	Institute for Electrical and Electronics Engineers
IG	Inspector General (Treasury)
IMP	Investment Management Process
IMS	Infromation Management System

INOMS	Integrated Network and Operations Management System
INS	Immigration and Naturalization Service
INSPASS	INS Passenger Accelerated Service System
IPS	Invoice Processing System
IRB	Investment Review Board
IRM	Information Resources Management
IRS	Internal Revenue Service
IS	Information Systems
ISLC	Information System Life Cycle (Treasury)
ISO	International Organization for Standardization
ISP	Information Systems Plan
IT	Information Technology
ITMRA	Information Technology Management Reform Act
JAD	Joint Application Development
JCL	Job Control Language
JFMIP	Joint Financial Management Improvement Program
JRP	Joint Requirements Planning
LAN	Local Area Network
LANIWAN	Local Area Network in a Wide Area Network
LEIS II	Law Enforcement Information System
LMS	Logistics Management System
LOC	Lines of Code
LTM	Legacy Transition Meta-Model
MIL-STD	Military Standard
MIPS	Million Instructions Per Second
MOA	Memorandums of Agreement
MOD-ACT	Customs Modernization and Informed Compliance Act
NADDIS	Narcotics and Dangerous Drugs Information System
NADIN	National Airspace Data Interchange Network NAILS
NAILS National	Automated Immigration Lookout System
NCAP	National Customs Automation Program
NCIC	National Crime Information Center
NIST	National Institute of Standards and Technology
NLECC	National Law Enforcement Communication Center
NLETS	National Law Enforcement Telecommunications System
NPR	National Performance Review
NPV	Net Present Value
NSA	National Security Agency
OAG	Official Airline Guide
ODTC	Office of Defense Trade Controls
OF	Office of Finance
OGA	Other Government Agencies
OHR	Office of Human Resources
OIM	Office of Information Management
OIT	Office of Information and Technology

OMB	Office of Management and Budget
O/S	Operating System
OSE	Open System Environment
OTAR	Over-The-Air-Rekeying
OTM	Office of Telecommunications Management
PASS	Passenger Accelerated Service System
PAT	Process Action Team
PAXLST	Passenger List
PC	Personal Computer
PCAS	Project Cost Accounting System
PCS	Purchase Card System
PEO	Program Executive Officer
PIMS	Property Information Management System
PIR	Post Implementation Review
PMT	Process Management Teams
PM	Program Manager
POC	Point of Contact
PRA	Paperwork Reduction Act (1980)
PRATS	Postage Request and Tracking System
PTO	Project Tracking and Oversight
PUB	Publication
QA	Quality Assurance
QAT	Quality Assurance Testing and/or Quality Action Team
RAAD	Rapid Architecture Application Development
RAC	Resident Agent in Charge
RAD	Rapid Application Development
RAMIS	Regulatory Audit Management Information System
RTC	Remote Terminal Controllers
SAC	Special Agent in Charge
SAT	System Acceptability Test
SC&RG	Security Compliance and Review Group
SCE	Software Capability Evaluation
SCM	Software Configuration Management
SDLC	System Development Life Cycle
SEACATS	Seized Asset and Case Tracking System
SEC	Securities and Exchange Commission
SED	Shippers Export Declarations
SEI	Software Engineering Institute
SEPG	Software Engineering Process Group
SLA	Service Level Agreement
SNA	System Network Architecture
SPA	Software Process Assessment
SQA	Software Quality Assurance
STC	Strategic Trade Centers
STD	Standard

TAFIM	Technical Architecture Framework for Information Management
TAG	Technical Architecture Group
TCP/IP	Transmission Control Protocol/Internet Protocol
TCS	Treasury Communications System
TDD	Teletype Device for the Deaf
TDP	Treasury Department Publication
TECS	Treasury Enforcement Communications System
TIERS	Treasury Information Executive Repository System
TII	Treasury Information Infrastructure
TIRB	Treasury IRB
TMS	Tape Management System
TRB	Technical Review Board
TSN	Trade Support Network
USMTF	Uniform Services Message Text Format
URD	User Requirements Document
USCS	United States Customs Service
VAN	Value Added Network
VHF	Very High Frequency Radio Voice Privacy Program
VM	Virtual Machine
VS	Virtual Storage
WAN	Wide Area Network
WWW	World Wide Web
Y2K	Year 2000